

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу**

До захисту допущено
В. о. завідувача кафедри
_____ О.Л. Тимощук
« ____ » _____ 20__ р.

Дипломна робота

**на здобуття ступеня бакалавра
за освітньо-професійною програмою «Системний аналіз і управління»
спеціальності 124 "Системний аналіз"
на тему: «Сервіс для проведення медичного діагностування на основі
машинного навчання»**

Виконав (-ла):

Студент (-ка) IV курсу, групи КА-61

Климчук Ярослав Ігорович _____

Керівник:

доцент, к.т.н. Дідковська М.В. _____

Консультант з економічного розділу:

доцент, к.е.н. Шевчук О.А. _____

Консультант з нормоконтролю:

доцент, к.т.н. Коваленко А. Є. _____

Рецензент:

к.т.н., доцент Т. М. Заболотня кафедра ПЗКС ФПМ _____

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 124 "Системний аналіз"

Освітньо-професійна програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри

_____ О.Л. Тимошук

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

Климчуку Ярославу Ігоровичу

1. Тема роботи «Сервіс для проведення медичного діагностування на основі машинного навчання», керівник роботи к.т.н., доцент Дідковська Марина Віталіївна, затверджені наказом по університету від «__» _____ 20__ р. № _____

2. Термін подання студентом роботи _____

3. Вихідні дані до роботи: модель класифікації зображень, система розпізнавання захворювання за медичним зображенням.

4. Зміст роботи: розробка системи медичного діагностування за вхідним зображенням

5. Перелік ілюстративного матеріалу: презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Шевчук О.А., доцент		

7. Дата видачі завдання: 13.04.2020 _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми ДР	13.04.2020 – 21.04.2020	виконано
2	Ознайомлення зі структурою БДР згідно з Положенням про державну атестацію студентів НТУУ «КПІ ім. І. Сікорського»	21.04.2020 – 28.04.2020	виконано
3	Ознайомлення з ДСТУ 3008-95 та стандарти ЄСПД	28.04.2020 – 05.05.2020	виконано
4	Проведення аналізу предметної області відповідно до теми БДР та узгодження результатів з керівником	05.05.2020 – 12.05.2020	виконано
5	Робота над базовою архітектурою програмного продукту	05.05.2020 – 22.05.2020	виконано
6	Розробка сервісу для програмного продукту	15.05.2020 – 22.05.2020	виконано
7	Аналіз отриманих результатів	22.05.2020 – 27.05.2020	виконано
8	Оформлення БДР	22.05.2020 – 27.05.2020	виконано

Студент

Климчук Ярослав Ігорович

Керівник

Дідковська Марина Віталіївна

РЕФЕРАТ

Дипломна робота: 129 с., 29 рис., 9 табл., 2 дод., 21 джерело.

SERVIS, DJANGO, КОНВОЛЮТИВНІ НЕЙРОННІ МЕРЕЖІ, МАШИННЕ НАВЧАННЯ, ДІАГНОСТУВАННЯ.

Предмет дослідження: системи проведення медичної діагностики шляхом класифікації зображень на факт наявності захворювання. Метод дослідження – конволютивні нейронні мережі, із модифікаціями для бінарної та мультикласової класифікації.

Мета роботи: спроектувати та реалізувати сервіс для проведення медичного діагностування за шляхом класифікації медичних зображень. Сервіс повинен діяти як готовий продукт, мати інтеграцію з базою даних.

Актуальність зумовлена практичною потребою покращення якості класифікації захворювань та автоматизації відповідного процесу прийняття рішень. Під час виконання роботи проведено аналіз існуючих алгоритмів класифікації зображень, обрано найбільш підходящий алгоритм. Створено сервіс, що є готовим програмним продуктом, із можливістю інтеграції до існуючої системи прийняття рішень у медичній сфері.

ABSTRACT

Bachelor's thesis: 129 p., 29 figs, 9 tab., 2 appendix, 21 sources.

SERVICE, DJANGO, CONVOLUTIONAL NEURAL NETWORKS, MACHINE LEARNING, DIAGNOSTICS

The theme of the work is the Service for conducting medical diagnosis based on machine learning.

Subject of research: medical diagnostics systems, convolutional neural networks, transformed for binary and multiclass classification.

The purpose of the work: create a service as a product-oriented model for conducting medical diagnosis on input pictures based on machine learning models. Service also should be integrated to database.

The actuality of this theme is supposed by practical needs for achieving better results of classification and automatization of current decision-making process in medical sphere.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1 Актуальність задачі проведення медичної діагностики.....	6
1.2 Аналіз існуючих підходів до вирішення задачі проведення медичного діагностування.....	8
1.2.1 Аналіз існуючих платформ	12
1.2.1.1 Buoy Health	12
1.2.1.2 Zebra Medical Vision.....	13
1.2.1.3 PathAI.....	14
1.2.1.4 BenevolentAI	14
1.3 Формалізація постановки задачі дослідження	15
1.4 Висновки до розділу 1	16
РОЗДІЛ 2 МАТЕМАТИЧНІ ОСНОВИ РОЗВ'ЯЗАННЯ ЗАДАЧІ ПРОВЕДЕННЯ МЕДИЧНОЇ ДІАГНОСТИКИ НА ОСНОВІ ВХІДНИХ ЗОБРАЖЕНЬ МЕТОДАМИ МАШИННОГО НАВЧАННЯ	17
2.1 Аналіз математичних основ розв'язання задачі.....	17
2.2 Методи машинного навчання для вирішення задачі медичної діагностики за вхідним зображенням	20
2.2.1. Використання конволютивних нейронних мереж (Convolutional Neural Networks)	20
2.2.2 Використання класичних методів машинного навчання.....	29
2.2.2.1 Лінійний класифікатор (Linear model)	31
2.2.2.2 Метод опорних векторів (Support Vector Machines).....	36
2.2.2.3 Дерева прийняття рішень (Decision Trees)	40
2.3 Порівняльний аналіз методів класифікації зображень.....	41
2.4 Критерії оцінки якості роботи моделі класифікації зображень	43
2.5 Метод проведення медичного діагностування захворювань	45
2.6 Висновки до розділу 2	46
РОЗДІЛ 3 АНАЛІЗ АРХІТЕКТУРИ ТА ІНТЕРФЕЙСУ ПРОГРАМНОГО ПРОДУКТУ.....	47

3.1 Обґрунтування вибору платформи та мови програмування.....	47
3.2 Аналіз вхідних даних та керівництво користувача	49
3.3 Аналіз архітектури програмного продукту	53
3.4 Опис інтерфейсу програмного продукту	58
3.5 Аналіз отриманих результатів	61
3.6 Висновки до розділу 3	64
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО – ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ.....	65
4.1 Обґрунтування функцій та параметрів програмного продукту	65
4.2 Економічний аналіз варіантів розробки ПП.....	70
4.3 Висновки до розділу 4	76
ВИСНОВКИ.....	77
ПЕРЕЛІК ПОСИЛАНЬ	79
ДОДАТОК А ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ.....	81
ДОДАТОК Б КОД ПРОГРАМНОГО ПРОДУКТУ	88

ВСТУП

Проведення медичної діагностики захворювань є першочерговою задачею сучасної медицини. Кожен із нас упродовж усього життя потребує якісних та своєчасних медичних послуг, одним із найважливіших аспектів яких є діагностування захворювань, адже правильний і своєчасний діагноз може мати великий вплив на життя і здоров'я людини.

Саме тому, сфера медицини була і залишається одним із найвагоміших об'єктів для досліджень. Із розвитком науково-технічної революції, відбувається бурхливий перехід до автоматизації процесів прийняття рішень у сфері медицини. У даній роботі буде розглянута задача проведення медичного діагностування, зокрема, буде реалізовано автоматизований процес прийняття рішень на прикладі діагностування пневмонії, крововиливу у мозок та цукрового діабету за вхідним зображенням відповідної частини тіла.

Об'єктом досліджень є методи класифікації зображень.

Предметом дослідження є автоматизовані системи прийняття рішень у сфері медицини.

Метою дослідження є створення сервісу – програмного продукту для проведення медичного діагностування на основі даного вхідного зображення методами машинного навчання у рамках автоматизації процесу прийняття рішень.

Актуальність даного дослідження полягає у тому, що стрімкий розвиток методів машинного навчання, водночас із збільшенням потреб автоматизації процесів у сучасній медицині, означає прискорення процесу симбіозу цих галузей.

Структурно, робота складається з чотирьох розділів.

У першому розділі буде детально розглянуто предмет дослідження, проаналізовано існуючі підходи та методи вирішення поставленої задачі, а також проведено їх порівняльну характеристику.

У другому розділі буде сформульовано основні принципи машинного навчання, розглянуто основні методи та підходи до автоматичної класифікації

зображень. Також будуть описані математичні основи та принципи практичної реалізації даних методів та проведено їх порівняльний аналіз.

У третьому розділі буде описано роботу сервісу, розглянуто архітектуру програмного продукту, шляхом побудови діаграми відповідних класів та таблиць в базі даних. Також буде проведений опис інтерфейсу користувача та здійснено аналіз основних висновків роботи програмного продукту.

У четвертому розділі буде проведено функціонально-вартісний аналіз роботи.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність задачі проведення медичної діагностики

На сьогоднішній день, методи машинного навчання все частіше застосовуються у найрізноманітніших сферах людської життєдіяльності, так як процеси прийняття рішень стають більш комплексними і взаємопов'язаними. Яскравим доведенням цього є той факт, що дана тематика розглядається навіть на державному та політичному рівні багатьох передових країн світу. У липні 2017 року Китай затвердив план розвитку штучного інтелекту наступного покоління, в березні 2018 року президент Франції Еммануель Макрон заявив про інвестування 1.5 мільярда євро в розвиток штучного інтелекту до 2022 року, у 2018 році Німеччина, Великобританія та Швеція створили власні аналоги для даної наукової сфери, а в квітні 2018 року Європейська комісія сформулювала єдину стратегію розвитку штучного інтелекту для країн Європи.

Розглядаючи сферу медицини, ми можемо констатувати той факт, що машинне навчання та штучний інтелект в цілому здійснили великий прорив у даній галузі: її застосування проглядається від медичного діагностування хворих до виробництва ліків зокрема. Говорячи про медицину, ми можемо підкреслити той факт, що ще не так давно усі знання та методи у даній сфері базувались на евристичних методах, із високим відсотком знань, що були отримані у процесі практичних підходів. Із плином часу, кількість і якість накопичених знань та інформації суттєво підвищилась, адже зросла потужність медичних приладів, що застосовуються для збору необхідної інформації про суб'єкт медичного дослідження. Таким чином, саме завдяки поліпшенню функціоналу обчислювальних ресурсів, внесенню змін до зберігання та обміну даними потенціал використання методів машинного навчання в медицині стрімко зріс.

На сьогоднішній день, радіологія, виявлення патологій та лабораторна медицина є прикладами технічних областей медичної галузі, для більшості задач яких уся необхідна інформація є структурованою та оцифрованою, а отже,

застосування методів штучного інтелекту, зокрема методів прогнозування, є логічно обґрунтованим для вирішення подібних задач. Серед актуальних прикладів можемо виділити такі:

- а) виявлення раку легень на основі даних комп'ютерної томографії;
- б) оцінка ймовірності наявності серцево-судинних захворювань на основі даних електрокардіограм та зображень томографії серця;
- в) класифікація наявності діабету на основі зображень сітчатки ока;
- г) оцінка вірогідності наявності пневмонії на основі рентгенівських зображень легень.

Зважаючи на велику кількість зібраних даних, моделі машинного навчання, побудовані на їх основі, можуть максимально точно опрацьовувати дану інформацію, виконуючи роль експерта. Втім, час, витрачений на встановлення діагнозу та прийняття рішення щодо подальшого лікування пацієнта, буде більш оптимізованим. Також зазначимо, що мова не йде про повноцінну заміну лікарів, а лише про автоматизацію процесів шляхом зменшення щільності потоку пацієнтів, швидше та якісніше виділяючи найбільш складні випадки захворювань.

Надалі наведемо більш конкретні приклади про впровадження штучного інтелекту у сферу медицини.

Виробництво ліків, як відомо – це дуже важкий та ресурсозатратний процес. Однак, із застосуванням алгоритмів машинного навчання, аналітичні процеси можна зробити більш ефективними, що у свою чергу скоротить час очікування випуску продукції та зменшить витрати на її виробництво. Так наприклад, у статті [1] зазначається, що термін розробки нових ліків усього два десятиліття тому займало приблизно 12 років, що у грошовому еквіваленті коштувало 1 мільярд доларів США. Натомість наразі з впровадженням методів машинного навчання процес розробки медичних препаратів вразі пришвидшився. Про це стверджує Шон Екінс та його колеги у статті [2], досліджуючи можливості застосування моделей до всіх аспектів прогнозування життєздатності молекули для подальшого клінічного використання. Наразі ж штучний інтелект успішно запроваджується на основних етапах виробництва медичних препаратів, а саме:

- а) розробки лікарських засобів;
- б) виявлення мети застосування;
- в) виявлення ключових елементів для ліків;
- г) пришвидшення клінічних випробувань;
- г) діагностика захворювання.

Розглядаючи задачу проведення медичного діагностування, за даними дослідження [3], ми можемо підсумувати те, що у надважких задачах, коли складність ознак може сильно впливати на точність та степінь адекватності отриманих результатів, методи машинного навчання, які базуються на класифікації зображень мають величезний потенціал. Таким чином, можливість застосування машинного навчання на великій кількості даних може значно пришвидшити процес прийняття рішень у сфері діагностики захворювань.

1.2 Аналіз існуючих підходів до вирішення задачі проведення медичного діагностування

Математично задачі проведення медичного діагностування приймають вигляд задач мультикласової класифікації, де за вхідними параметрами — симптомами пацієнта — визначається тип хвороби.

Загалом для вирішення такого плану задач існує три базових підходи:

- а) статистичний підхід;
- б) логічний підхід;
- в) біонічний підхід.

Надалі надамо більш детальну характеристику для кожного із зазначених вище способів одержання результатів діагностування.

На сьогоднішній день з точки зору легкості інтерпретації найпоширенішим підходом прийняття рішень під час проведення медичного огляду пацієнта є логістичні методи. За своїм принципом дії вони повторюють ланцюг міркувань

самого експерта-лікаря, адже так само мають бути чітко визначеними, послідовними та обґрунтованими.

Одним із прикладів логістичного підходу є алгоритм машинного навчання – дерева рішень. Теза про послідовність базується на тому, що даний метод по суті є ієрархічною структурою правил, які застосовуються у відповідній послідовності. Початок алгоритму стартує з предикату у найвищому вузлі дерева, який також ще називають його коренем, і закінчується у вузлі з останнім питанням. Усі правила будуються за структурою «якщо..., то...» і розбивають вхідну множину елементів – симптомів пацієнта – на дві підмножини. Цей процес повторюється рекурсивно до тих пір, поки не буде утворено листка на дереві. Інакше кажучи процес обробки симптомів триватиме до моменту такого розподілу ознак досліджуваного об'єкта (пацієнта), за набором яких можна буде встановити остаточний діагноз.

Також варто зазначити, що, як правило, симптоми не мають рівнозначної важливості для алгоритму. У цьому і полягає характеристика того, що відбір ознак досліджуваного об'єкта, тобто симптомів пацієнта, та формування правил на їх основі мають бути чітко визначеними та обґрунтованими.

Таким чином основними діями алгоритму є розмітка важливості ознак та визначення послідовності перевірки правил.

Наступним підходом до вирішення задачі проведення медичного діагностування розглянемо статистичні методи. Представниками даного інструментарію є метод дискримінантного аналізу, баєсовський класифікатор, вивід на основі прецедентів та метод фазового інтервалу.

Дискримінантний аналіз зводиться до розробки вирішення задач класифікації шляхом розрізнення об'єктів спостереження (пацієнтів) за відповідними ознаками. Точність класифікації за даного аналізу визначається як частка правильно класифікованих об'єктів за допомогою прогностичного рівняння спостережень. За умов коли початкова вибірка даних має велику розмірність, тоді аналіз проводиться лише для частини даних. Для іншої половини даних класифікація об'єктів спостережень відбувається за прогностичним рівнянням, побудованим за екземплярами першої частини.

Так само як і для логістичних методів, під час дискримінантного аналізу також проводиться вибірка ознак, які братимуть участь у процесі прогнозування, з усієї вихідної множини.

Використання баєсовського класифікатора для задач медичної діагностики пацієнтів пояснюється досить простою побудовою моделі, нескладними математичними розрахунками та легкою інтерпретованістю результатів. Втім варто зазначити, що даний метод погано відловлює нетрадиційні симптоми певної хвороби, що відповідно зменшує точність ідентифікації захворюваності. Окрім цього необхідною передумовою для застосування даного класифікатора є велика база архівних даних щодо випадків захворюваності пацієнтів. Це потрібно задля того, аби результат встановленого діагнозу відповідав дійсності.

Вивід на основі прецедентів, як ще один представник статистичного підходу діагностування захворюваності, базується на прикладах попередніх випадків і досвіду. Прийняття рішень за його використання відбувається в такій послідовності. Спочатку здійснюється пошук схожих прецедентів (симптомних випадків) у минулому, а потім розраховується міра близькості між новим і всіма знайденими аналогічними випадками, які раніше вже були співставленні з певною хворобою.

Втім така прив'язка до попереднього досвіду призводить до двох основних недоліків. По-перше, база попередніх випадків та загального досвіду експертів є досить суб'єктивною та не має узагальнюючої здатності.

А по-друге, для пацієнтів з новими формами хвороби за застосуванням методу виводу на основі прецедентів не можна буде визначити діагноз, адже бази захворювань для таких симптомів просто не існуватиме на той момент. Тобто все вказує на те, що даний метод працює лиш для часткових випадків, а саме для таких, коли пацієнт має типові симптоми хвороби, а сама хвороба є загально відомою.

За методом фазового інтервалу функціонування організму людини можна записати у бінарному вигляді, коли нуль відповідає за відсутність певної ознаки, тобто симптому пацієнта, а одиниці навпаки – за її наявність. За таких умов результат стану здоров'я пацієнта перетворюється на точку фазового простору. Симптоми одного типу хвороби групуються, утворюючи область точок на просторі.

Для таких груп визначаються координати центру області та її діаметр. Центром області вважають точку, яка відповідає найбільш поширеному набору ознак стану здоров'я для певного типу захворювання, а діаметром – максимальний розмах між двома точками області.

Встановлення діагнозу за даним методом проводиться за розрахунком мінімальної відстані від центрів областей (симптомів хвороб) до наявного набору симптомів нового пацієнта.

Також варто зазначити, що за даного методу кожен стан здоров'я пацієнта (точка у фазовому просторі) має свою вагу важливості, яка надалі впливає на розрахунок міри відстані. Введення вагових коефіцієнтів з одного боку приводить до покращення точності прогнозування діагнозу, втім з іншого робить цей метод більш складним, адже для кожного типу захворюваності, що перевірятиметься за даним методом, потрібно переналаштовувати вектор ваг.

Останнім підходом до встановлення діагнозу пацієнтів є біонічна група методів. До даної групи методів належать ті методи, математичний апарат яких заснований на прикладі роботи людського мозку. Серед найяскравіших представників даних методів виділяються нейронні мережі.

Нейромережі використовуються для вирішення основного спектру задач медичної діагностики: задач класифікації та кластеризації захворювань, прогнозування різноманітних епідеміологічних факторів. Дана популярність зумовлена тим, що нейромережі можуть бути використані не тільки як алгоритм для прийняття рішень по тому чи іншому пацієнту, а і як застосування для збору додаткових ознак із неструктурованих даних, таких як рентгенівські знімки чи записи кардіограми роботи серця.

Генерація нових ознак для подальшого прийняття рішення при проведенні медичної діагностики відбувається за рахунок інваріантності нейромереж до вхідних зображень та відео, що дає можливість до їх використання на великих масивах даних без ітеративного оновлення.

Серед недоліків даного підходу слід виділити те, що для навчання нейромереж потрібні великі масиви даних та обчислювальні ресурси, але з

розвитком хмарних технологій, зокрема архітектур збереження та обробки великих масивів даних, даний недолік стає все менш і менш помітним при практичному застосуванні.

1.2.1 Аналіз існуючих платформ

Застосування методів машинного навчання в медичній сфері не є новизною, тому й не дивно що на сьогоднішній день вже існує різноманітна кількість продуктів такої кооперації. Насамперед мова йде про розробку програмних продуктів на базі штучного інтелекту, які стали незамінним помічником як для медичного персоналу, так і для пацієнтів водночас. Розглянемо наступний перелік програмних додатків:

- а) Buoy Health;
- б) Zebra Medical Vision;
- в) PathAI;
- г) BelevolentAI.

1.2.1.1 Buoy Health

Buoy Health – це застосунок для діагностики симптомів хворого та проведення його лікування, що діє за принципом роботи чат-боту. Архітектурно, принцип роботи даного чат-боту базується на використанні Dialog Flow – платформи, що дозволяє використовувати методи машинного навчання для обробки природної мови (natural language processing).

Спілкування з пацієнтом побудоване таким чином, щоб максимально прискорити процес прийняття рішення, шляхом оптимального підбору запитань (див. рис. 1.1). Таким чином, процес визначення діагнозу полягає у проходженні

пацієнтом опитування, після якого відбувається скринінг результатів, за висновками якого можливі такі випадки:

- а) надання інструкцій щодо подальшого лікування;
- б) створення запису до компетентного медичного працівника у найближчій для пацієнта лікарні.

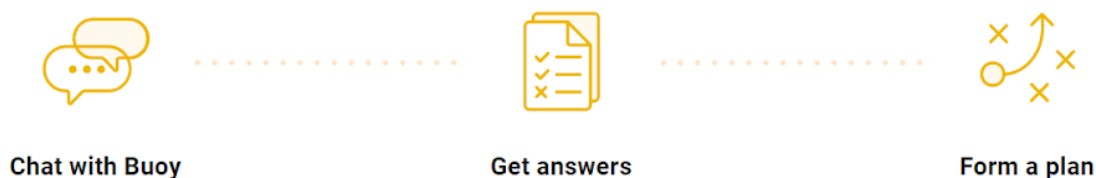


Рисунок 1.1 – Принцип дії чат-боту Buoy Health або [4]

1.2.1.2 Zebra Medical Vision

Zebra Medical Vision – це сервіс, розроблений із застосуванням алгоритмів глибинного навчання (deep learning), зокрема глибинних нейронних мереж (deep neural networks) для асистування лікарів. Дана аналітична система на вхід отримує відскановані зображення захворювань головного мозку, легень, печінки, серцево-судинної системи, кісток і проводить автоматичний аналіз даних зображень задля пошуку клінічного захворювання.

Даний сервіс надає послуги у режимі реального часу, допомагаючи радіологам визначити правильний діагноз вже на етапі проведення ранньої діагностики. На даний момент у компанії діє 48 алгоритмів машинного навчання для різноманітних захворювань. Деякі з них вже пройшли апробацію у Європі та США, що говорить про високу предикативну здатність алгоритмів, що там використовуються.

Щодо перспектив розвитку, то можна зазначити, що дана компанія планує створити інструмент для класифікації рентгенівських знімків грудної клітини з метою ідентифікації можливих проявів туберкульозу.

1.2.1.3 PathAI

PathAI – це компанія, що пропонує сервіс для надання допомоги лікарям при діагностуванні ракових пухлин, а також займається персоналізацією процесу надання медичних послуг шляхом використання методів машинного навчання.

Ще одна місія програмного продукту полягає у наданні допомоги лікарям у передбаченні реакції пацієнта на проведену терапію. Даний прогноз базуватиметься на характеристиках тканин шкіри хворих.

Щодо підходів, які використовуються цією платформою, можна виділити цілий ряд методів навчання з підкріпленням, які використовуються в якості алгоритмів персоналізації. Розробка даного продукту є прикладом реалізації саме алгоритмів машинного навчання з підкріпленням у середовищі реалізації продукту (production).

1.2.1.4 BenevolentAI

BenevolentAI – це сервіс для проведення персоніфікації процесу прийняття рішень під час проведення медичного діагностування. Під процесом персоніфікації будемо розуміти надання максимально коректного медичного втручання, з умовою дотримання коректності часу даного втручання. У той же час, даний продукт активно займається і розробкою ліків. Зокрема, BenevolentAI використовує методи глибинного навчання для аналізу з'єднань молекул, які могли б принести максимальний результат при боротьбі з тим чи іншим захворюванням.

На сьогоднішній день дослідники цієї компанії беруть активну участь у розробці препарату, клінічні тести якого підтвердили його значимість при боротьбі з

COVID-19. Попередньо даний препарат (baricitinib) застосовували для лікування ревматоїдного артриту.

Наведемо порівняльну характеристику зазначених програмних продуктів (табл. 1.1):

Таблиця 1.1 – Порівняльна характеристика програмних продуктів на базі штучного інтелекту

Критерій порівняння	Buoy Health	Zebra Medical Vision	PathAI	BenevolentAI
Наявність зовнішніх залежностей	+	-	-	-
Наявність міжнародних сертифікатів якості	-	+	-	+
Інтегрованість в існуючі медичні системи	-	+	-	+

Джерело: розроблено автором

Як бачимо, на даний момент Zebra Medical Vision та BenevolentAI вже перейшли стадію від стартапа до продуктової компанії, продукти якої використовуються в існуючих медичних системах провідних країн світу, що говорить про величезний потенціал застосування методів машинного навчання у медицині.

1.3 Формалізація постановки задачі дослідження

Проаналізувавши предметну область та вже існуючі платформи в даній області, можна провести таку формалізацію постановки задачі:

а) провести дослідження підходів до вирішення задачі проведення медичного діагностування шляхом класифікації зображень;

б) розробити власний алгоритм вирішення даної задачі;

- в) провалідувати отримані результати;
- г) побудувати сервіс, який буде використовувати побудований алгоритм.

1.4 Висновки до розділу 1

У даному розділі було розглянуто процес діагностування пацієнтів в медичних системах світу та показана його актуальність. Наступний етап полягав у проведенні аналізу існуючих програмних продуктів, визначенні їх основних структурних та архітектурних елементів, а також виявленні їх впливу на сьогоденну структуру медичної сфери. У результаті цього можна сформулювати такі основні напрямки:

- а) створення персоналізованих систем прийняття рішень у медичній сфері;
- б) розробка лікарських препаратів;
- в) здійснення класифікації захворювань із використанням різних вхідних даних.

Крім цього, у першому розділі було з'ясовано степінь бізнес розвитку компаній-розробників та проведено порівняльну характеристику відповідних продуктів.

Також окремим пунктом даного розділу є формалізація постановки задачі предметної області.

РОЗДІЛ 2 МАТЕМАТИЧНІ ОСНОВИ РОЗВ'ЯЗАННЯ ЗАДАЧІ ПРОВЕДЕННЯ МЕДИЧНОЇ ДІАГНОСТИКИ НА ОСНОВІ ВХІДНИХ ЗОБРАЖЕНЬ МЕТОДАМИ МАШИННОГО НАВЧАННЯ

2.1 Аналіз математичних основ розв'язання задачі

Сформалізувавши у попередньому розділі задачу дослідження, перейдемо до математичних основ розв'язання задачі. Поставлена проблема є однією з найпоширеніших та найпопулярніших задач, що розв'язуються методами машинного навчання з учителем – задачі класифікації.

Машинне навчання – розділ штучного інтелекту, що займається побудовою та вивченням методів, які здатні навчатися. Під “навчанням” методу у більшості випадків розуміють навчання за прецедентами, загальне формулювання якого полягає у наступному: по заданій скінченній множині об'єктів (прецедентів), які описуються деяким набором даних, що зібрані по кожному із об'єктів множини, потрібно виявити та відновити певні закономірності та взаємозв'язки. Сукупність таких об'єктів, із заданим характеристичним описом, часто називають навчальною вибіркою. Під час навчання за прецедентами фіксується функціонал якості – критерій, згідно якого буде проходити процес навчання і значення якого показує, наскільки добре побудована модель описує вхідні дані, та деяка сім'я моделей, з якої буде обиратись фінальний алгоритм – побудована модель. Сім'я моделей має свою варіативність, що задається різним набором параметрів моделі. Фінальний алгоритм, обраний у процесі навчання, буде приймати оптимальне значення функціоналу якості на навчальній вибірці.

Фактично, кожна задача навчання алгоритму машинного навчання зводиться до задачі оптимізації (як правило, мінімізації) деякого функціоналу якості на множині прецедентів. Тобто, в результаті навчання алгоритму будуть підібрані такі параметри моделі машинного навчання, при яких обраний функціонал якості буде приймати оптимальне значення на множині об'єктів навчальної вибірки:

$$\vec{\alpha} = \operatorname{argmin}_{\vec{\alpha}} Q(X, \vec{\alpha}),$$

де $\vec{\alpha}$ – оцінюваний вектор гіперпараметрів моделі;

Q – обраний функціонал якості;

X – задана навчальна вибірка.

Говорячи про типи задач навчання за прецедентами, розглядають такі основні їх види:

- а) навчання з учителем (supervised learning);
- б) навчання без учителя (unsupervised learning);
- в) навчання з підкріпленням (reinforcement learning);
- г) часткове навчання (semi-supervised learning).

Розглянемо підвид задач машинного навчання, що використовується для вирішення поставленої задачі у роботі – навчання з учителем (див. рис. 2.1).

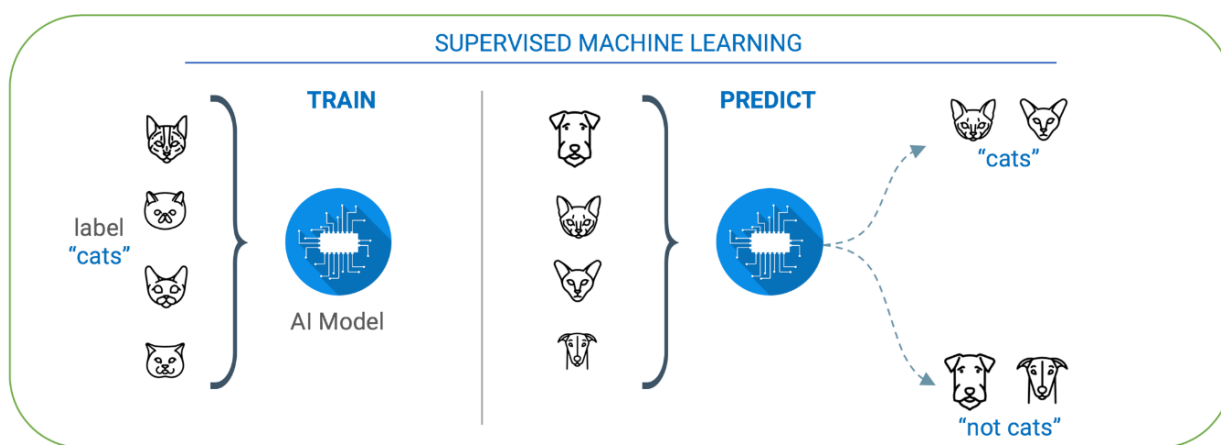


Рисунок 2.1 – Робота методу машинного навчання з учителем або [5]

Навчання з учителем характерне тим, що тренувальна вибірка, окрім об'єктів та їх характеристичних описів містить ще значення цільової змінної для кожного об'єкта. Таким чином навчання моделей машинного навчання з учителем полягає у тому, щоб побудувати такий алгоритм, який зможе генералізувати зв'язок між характеристичним описом об'єкта та цільовою змінною.

Фактично, моделі машинного навчання з учителем є дискримінативними, тобто такими, що моделюють умовну ймовірність отримання певного значення

цільової змінної, маючи заданий характеристичний опис певного об'єкта, і використовують змодельовану ймовірність у процесі прийняття рішень:

$$p = P(y \vee \vec{x}),$$

де p – ймовірність, що моделюється дискримінативними моделями;
 y – відповідне значення цільової змінної;
 \vec{x} – вектор описових характеристик об'єкта.

Розглядаючи основні види задач машинного навчання з учителем, насамперед виділяють такі:

- а) задачі класифікації;
- б) задачі регресії.

Ключовою відмінністю даних видів задач є тип цільової змінної: у задачах класифікації він скінченний, а в задачах регресії – ні.

Означимо загальне формулювання задачі класифікації – побудова алгоритму, який за характеристичним описанням об'єкта буде моделювати мітку класу, до якого належить даний об'єкт і тим самим розмічати отримані вхідні дані за їх описовими характеристиками.

Таким чином, задачу проведення медичної діагностики за заданим зображенням можна віднести до задачі класифікації зображень.

Розглянемо основні методи машинного навчання, що можуть бути використані для вирішення даної задачі.

2.2 Методи машинного навчання для вирішення задачі медичної діагностики за вхідним зображенням

2.2.1. Використання конволютивних нейронних мереж (Convolutional Neural Networks)

Нейронна мережа – це математична модель, що була сконструйована на основі принципів роботи людського нейрона. Звідси походять і основні складові архітектури класичної нейронної мережі: нейрони, синапси та функції активації нейронів.

Розглянемо детальніше кожен із основних архітектурних елементів класичної нейронної мережі.

Нейрон – основна структурна одиниця нейронної мережі, що отримує на вхід дані, трансформує їх та віддає на вихід. Як правило, нейрон може бути представлений у вигляді деякої нелінійної функції від лінійної комбінації усіх вхідних сигналів:

$$u = \varphi \left(\sum_{i=1}^n \alpha_i * x_i + b_i \right),$$

де u – вихідний сигнал нейрона;

φ – нелінійна функція активації нейрона;

n – кількість вхідних сигналів, що подаються на вхід нейрона;

α_i – ваги синапсів – зв'язків нейрона із попереднім шаром;

x_i – вхідні сигнали, що подаються на нейрон;

b_i – зсув (bias) нейрона.

Слід розглянути також і функції активації. Функція активації – функція, що використовується для зміни стану нейрона, і фактично визначає його вихідний сигнал. Дана функція слугує для того, щоб корегувати відклик нейрона на відповідний вхідний сигнал. Таким чином, в залежності від функції активації,

нейрон буде або активуватись на вхідний сигнал і формувати відклик, або ігнорувати вхідний сигнал, приймаючи інше значення.

Говорячи про функції активації, слід розуміти, що кожна з них має свої переваги і недоліки, та може бути використана в залежності від тої чи іншої задачі, що ставиться перед побудовою нейромережі. Розглянемо основні, найбільш часто вживані функції активації нейронів у нейронних мережах.

Однією з таких є ступінчата функція активації, що за своєю суттю є кусково-лінійною і визначається наступним чином:

$$\varphi(x, \alpha) = \begin{cases} 1, & x > \alpha \\ 0, & \text{в протилежному випадку} \end{cases}$$

Зокрема, частковим випадком даної функції є функція Хевісайда:

$$\mu(x) = \begin{cases} 1, & x > 0 \\ 0, & \text{в протилежному випадку} \end{cases}$$

Таким чином, при використанні даної функції, активований нейрон буде приймати значення 1, а неактивований – 0.

Дана функція активації знайшла досить широке використання у задачах бінарної класифікації – підвиду задач класифікації, де кількість класів, на які потрібно класифікувати об'єкт дорівнює двом. Водночас, така побудова функції унеможливорює її використання у задачах мультикласової класифікації, тобто коли кількість класів більше двох. Це обумовлено тим, що з використанням ступінчатої функції активації при мультикласовій класифікації, нейрон може бути активованим одночасно на декілька класів.

Наступним видом функції активації розглянемо сигмоїдну функцію активації, що визначається наступним чином:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

Сигмоїда – монотонно зростаюча, диференційована нелінійна функція. Дана функція є однією з найпопулярніших функцій активації і широко використовується в задачах класифікації. Так як ця функція не є бінарною, і приймає значення від 0 до 1, то вона часто використовується для моделювання ймовірності отримання певного класу в задачах класифікації.

Ще одним видом функції активації є гіперболічний тангенс.

За визначенням, гіперболічний тангенс визначається наступним чином:

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{2}{1 + e^{-2x}} - 1 = 2 * \text{sigmoid}(2x) - 1$$

Гіперболічний тангенс – ще одна часто вживана функція активації, що за властивостями подібна до сигмоїди і, фактично, є скорегованою сигмоїдною функцією. Значимою відмінністю даної активаційної функції від сигмоїди є те, що область значень гіперболічного тангенса – від -1 до 1, тому, на відміну від сигмоїдної функції активації, гіперболічний тангенс можна використовувати, коли припускається, що нейрон може мати від’ємні значення після активації. Водночас, якщо активаційні значення нейронів можуть бути тільки невід’ємними, то зазвичай надається перевага сигмоїдній функції.

І на останок розглянемо ReLu – функція активація, яку математично можна виразити наступним чином:

$$\text{ReLu}(x) = \max(0, x)$$

Ця активаційна функція особлива тим, що є універсальним апроксиматором, так як будь яка функція може бути апроксимована комбінацією ReLu. Вона стала дуже популярною з розвитком методів глибинного навчання, так як виконує більш прості математичні операції, і, як наслідок, потребує значно менше обчислювальних ресурсів порівняно з сигмоїдою чи гіперболічним тангенсом. Говорячи про методи

глибинного навчання, дана функція активації може бути використана і для того, щоб зробити активаційний шар нейронів більш розрідженим, тобто зробити так, щоб деякі нейрони не були активованими. Даний підхід широко застосовується у якості регуляризації моделей глибинного навчання, що дозволяє зробити їх більш робастними і, як наслідок, збільшити узагальнюючу здатність таких моделей.

Класична нейронна мережа складається із сукупності цих складових і утворює структуру, де кожен прихований шар складається із множини нейронів, кожен з яких має зв'язок з усіма нейронами попереднього шару.

Перший шар нейронів називають «вхідним» шаром, кількість нейронів в якому залежить від розмірності вхідних даних, останній повнозв'язний шар нейронів (тобто такий шар, де кожен нейрон наступного шару зв'язаний з кожним нейроном попереднього шару) називається вихідним шаром і у задачах класифікації він демонструє кількість класів, на яку ми хочемо «розбити» вхідну множину значень (див. рис. 2.2).

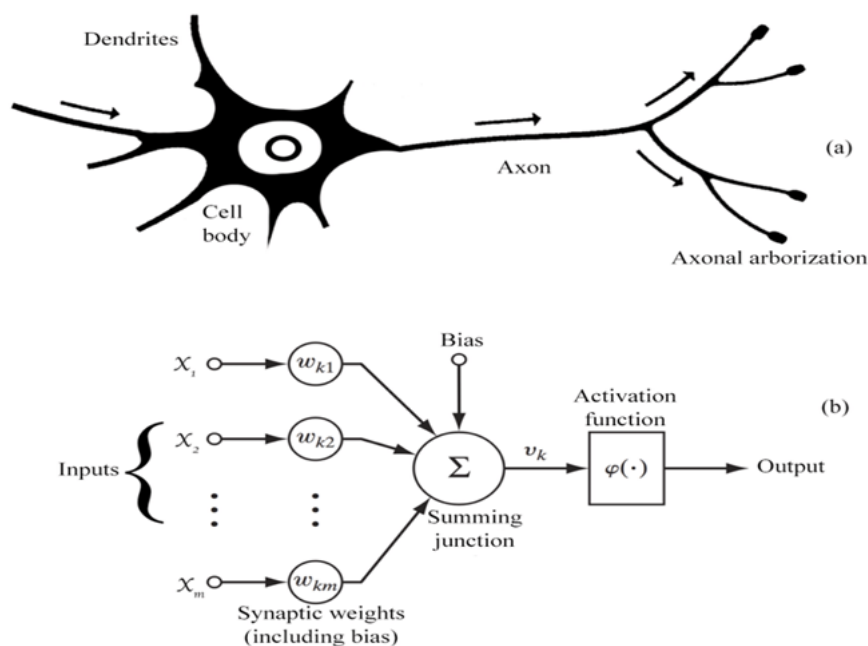


Рисунок 2.2 – Базова архітектура штучної нейронної мережі у порівнянні з людським нейроном або [6]

Слід також розглянути те, як проходить навчання нейромережі. Так як нейронні мережі є представником методів машинного навчання за прецедентами, то задача навчання зводиться до задачі мінімізації функціоналу помилки мережі.

Ключовою відмінністю навчання нейромереж від інших методів машинного навчання є те, що цей процес відбувається у два етапи:

- а) етап прямого розповсюдження помилки (forward propagation);
- б) етап зворотного розповсюдження помилки (back propagation).

Розглянемо більш детально кожен із етапів.

У процесі прямого розповсюдження помилки нейромережа розраховує прогнозний відклик моделі, шляхом поширення вхідного сигналу через всю архітектуру мережі.

Далі, у процесі зворотного розповсюдження помилки, розраховується вклад кожного нейрона у загальне значення обраного функціоналу якості і тим самим відбувається корегування кожного із вагових коефіцієнтів нейронів (див. рис. 2.3, 2.4).

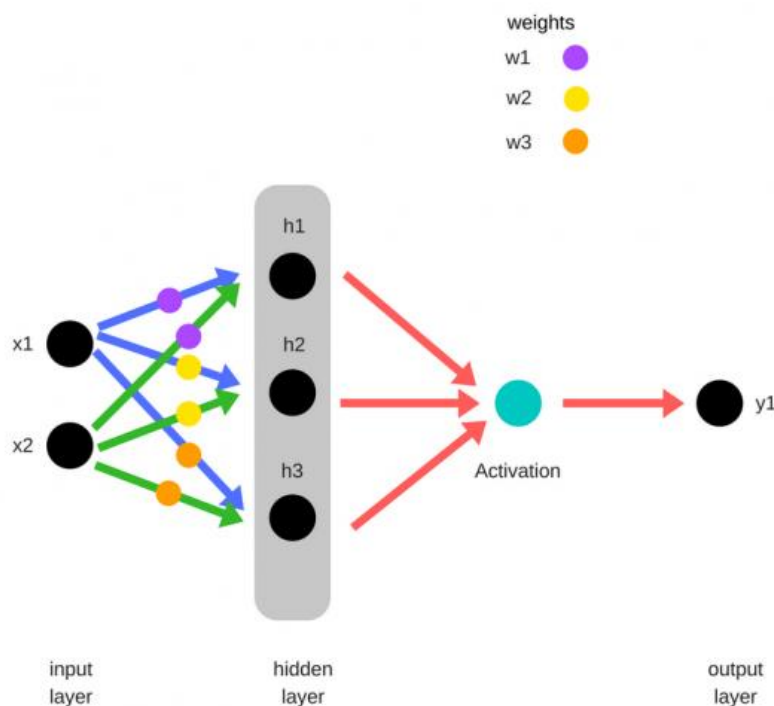


Рисунок 2.3 – Алгоритм прямого розповсюдження помилки або [7]

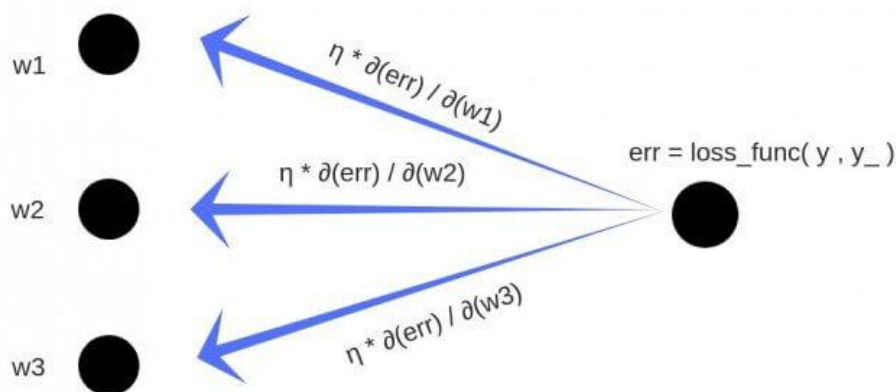


Рисунок 2.4 – Алгоритм зворотного розповсюдження помилки або [7]

У процесі зворотного розповсюдження помилки розраховується значення функції помилки на даному об'єкті, і відображається вклад кожного вагового коефіцієнта шляхом знаходження частинної похідної функції помилки по відповідному ваговому коефіцієнту. Фактично, процес навчання нейромережі зводиться до послідовного застосування методу градієнтного спуску на кожному шарі нейронної мережі.

Зауважимо, що під час проходження даного етапу навчання нейромережі, широко застосовується правило диференціювання складної функції (chain rule):

$$y = f(u(x)) \rightarrow \frac{dy}{dx} = \frac{dy}{du} \frac{du}{dx}$$

Дані етапи навчання нейронної мережі проводяться ітеративно, певну кількість епох – циклів виконання алгоритму навчання, і, як результат, отримуються оптимальні значення вагових коефіцієнтів - гіперпараметрів нейромережі.

Використання класичних нейромереж при розв'язанні задачі класифікації зображень було досить поширеним, проте, з розвитком методів глибокого машинного навчання, з'явився більш спеціалізований метод, що використовується для класифікації зображень – конволютивні нейронні мережі [8].

Конволютивні (згорткові) нейронні мережі (Convolutional Neural Network, CNN) – вид нейронних мереж, основним застосуванням яких є вирішення задач

класифікації та сегментації зображень. Так як ці мережі є удосконаленням звичайних нейромереж, то вони мають деякі модифікації у своїй архітектурі.

Основними шарами таких нейронних мереж є: шар згортки (Convolutional Layer), шар субдискретизації (Pooling Layer), шар Dropout та «звичайний» повнозв'язний шар (Dense Layer) нейронів.

Розглянемо детальніше кожен із основних архітектурних шарів конволютивних мереж.

Основою усієї нейромережі є шар «згортки» (convolution layer) і складається з «ядра згортки» – двовимірної матриці (частіше всього – розміром 3x3 або 5x5).

Процес «згортки» відбувається у такі етапи:

- а) вирахування скалярного добутку між ядром згортки та декодованою п-вимірною матрицею, що описує вхідне зображення;
- б) «зсув» ядра згортки на відповідний крок так, щоб вирахувати її скалярний добуток з тією частиною матриці вхідного зображення, яка не була використана на попередньому кроці.

Отже, упродовж перших двох етапів нейромережа сформує так звану «карту ознак» («feature map»), кількість цих карт залежить від кількості ядер згортки, кожне з яких описує певну ознаку, яка буде «відшукуватись» на вхідному зображенні (див. рис. 2.5).

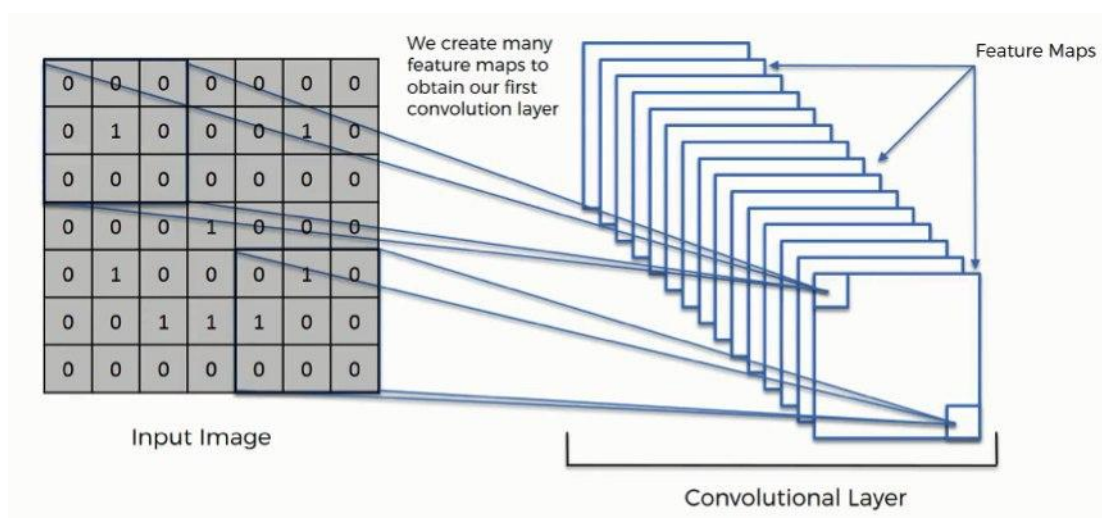


Рисунок 2.5 «Візуалізація роботи першого шару згортки розміром 5 x 5» або

На інших етапах згортки у якості вхідного зображення буде використовуватись та карта ознак, що була отримана на попередньому кроці.

Таким чином, на кожному наступному етапі згортки нейромережа буде вивчати все більш високоабстрактні закономірності, поки, врешті-решт, не зможе відрізнити такі поняття, як колір очей та форму обличчя (див. рис. 2.6).

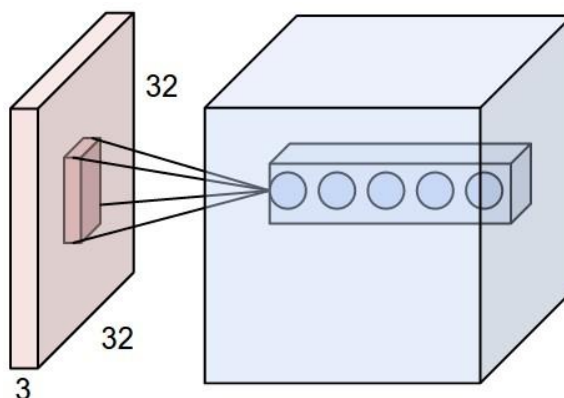


Рисунок 2.6 – Обробка кольорового зображення (RGB) розміром 32 x 32 вхідним шаром нейронів конволютивної нейронної мережі або [10]

Наступним розглянемо шар субдискретизації (Pooling layer).

Цей шар використовується для зменшення розмірності карти ознак, яку ми отримали після шару згортки. Це робиться для того, щоб перешкодити перенаванчання нейромережі, а також вивченню нелокальних закономірностей, адже для подальшого вивчення зображення нам вже не потрібно знати точне місцеположення тієї чи іншої ознаки, важливим залишається лише факт її наявності на даній ділянці зображення.

Реалізація цього шару є досить простою: ми проходимося по карті ознак спеціальним «віконечком» - двовимірною матрицею (частіше всього – розміром 2x2), тобто на кожному кроці розглядаємо лише 4 значення, які знаходяться поруч на карті ознак, і вибираємо максимальне з них (через дану особливість цей шар ще називають max-pooling).

Досить часто використовувався й інший спосіб: average pooling - коли ми на кожному кроці проходимося по карті ознак двовимірною матрицею і беремо середнє значення від ознак, які потрапили у відповідне «віконце». Але на практиці даний метод показує дещо гірші результати ніж max-pooling.

Таким чином, на виході цього шару, ми отримаємо ту ж саму двовимірну карту ознак, але її розміри по ширині та довжині будуть менші від вхідних у 2 рази.

Отже, чергуючи шари згортки та пулінгу ми можемо виділити із нашого вхідного зображення основні ознаки та сформувати з них відповідні вектори ознак, які ще називають дескрипторами.

Принцип роботи наступного шару нейронної мережі – шар Dropout – полягає у тому, що на цьому кроці ми випадково «вимикаємо» певний відсоток нейронів (зазвичай береться 20%), тобто робимо так, що дані нейрони не реагують на ознаки, що є виходом попереднього шару. Цей метод використовується для того, щоб запобігти перенавчанню нейронної мережі та зменшити акцентування «уваги» нейронів на незначних, нетипових ознаках. Зазвичай даний шар використовують після Pooling шару.

Вихідним шаром в архітектурі конволютивної нейронної мережі є повнозв'язний шар нейронів. Він використовується для обробки отриманих ознак із згорткових шарів.

Інакше кажучи з використання декількох шарів згортки, пулінгу та Dropout ми отримуємо багатовимірний вектор-стовпчик, кожна координата якого буде відповідати за деяку ознаку, що присутня на вхідному зображенні (це можуть бути певні градієнти кольорів, наявність паралельних ліній тощо). Далі, щоб власне класифікувати вхідне зображення (наприклад, визначити, котик це чи собака, або чи одна й та сама людина представлена на зображенні) потрібно використати повнозв'язний шар нейронів, який працює як «звичайна» нейронна мережа: на вхід кожного нейрона подається лінійна комбінація вагів та вхідних даних (у нашому випадку – ознаки, що отримані зі згорткових шарів), виходом нейрона буде та ж лінійна комбінація, «пропущена» через функцію активації даного нейрона.

Розмірність даного повнозв'язного шару (кількість нейронів у ньому) відповідає кількості класів, на які ми хочемо класифікувати вхідне зображення (у задачі про котиків та собак – 2).

Фактично, виходом k -го нейрона вихідного шару буде ймовірність того, що на вхідному зображенні представлено об'єкт k -го класу.

Приклад архітектури нейронної мережі показано на рисунку 2.7.

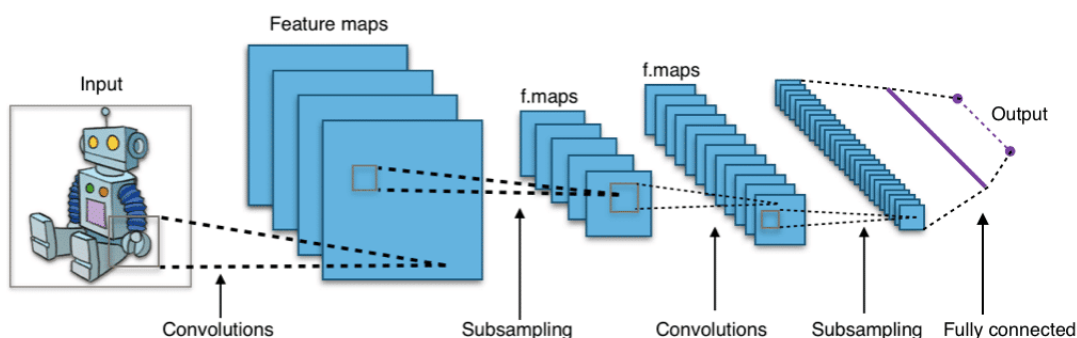


Рисунок 2.7 – Приклад CNN мережі [11]

Дана мережа приймає чорно-біле вхідне зображення розміром 32 x 32 та складається з двох шарів згортки (розмірностями 6 x 28 x 28 та 16 x 10 x 10 відповідно), двох шарів пулінга, а також трьох повнозв'язних шарів нейронів (розмірностями 120, 84, та 10 відповідно).

Розмірність останнього повнозв'язного шару означає, що дана нейронна мережа розв'язує задачу класифікації зображень на 10 класів (які, звичайно, не перетинаються)

Отже, даний метод полягає у побудові спеціалізованої нейронної мережі для класифікації зображень [12].

2.2.2 Використання класичних методів машинного навчання

Класичні методи машинного навчання, що використовуються для вирішення задач класифікації є типовими прикладами моделей, побудованих за принципом навчання за прецедентами: такі алгоритми використовують тренувальну вибірку з описовою характеристикою об'єктів та відомими значеннями цільової змінної.

Тобто для того, щоб застосувати класичний метод класифікації, потрібно спочатку отримати описову характеристику об'єктів – зображень. Можливими

способами застосування класичних методів машинного навчання для вирішення задач класифікації зображень є використання цих методів у зв'язці з іншими методами машинного навчання, що можуть бути використані на етапі генерування ознак при побудові так званих ембедингів – векторних представлень об'єктів, у даному випадку – зображень.

Даний підхід є типовим прикладом трансферного навчання – методу машинного навчання, який використовує більш спеціалізовані методи машинного навчання як стартову точку для застосування класичних методів машинного навчання (див. рис.2.8).

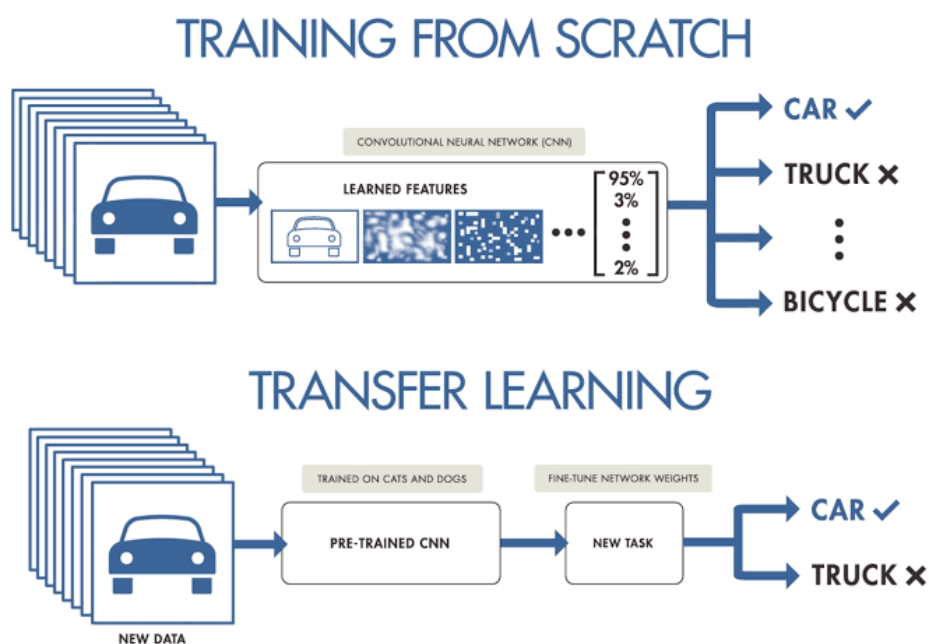


Рисунок 2.8 – Ілюстрація відмінностей між класичним та трансферним навчанням або [13]

Таким чином, побудова фінального алгоритму методом трансферного навчання проходить у два етапи:

- а) генерація ознак за допомогою спеціалізованої моделі машинного навчання, що навчалась для розв'язання іншої задачі;
- б) використання фінальним класифікатором отриманих ознак у якості вхідних характеристик, що описують об'єкт.

У даному випадку в якості спеціалізованого методу, задача якого - генерація описових характеристик зображень - виступає конволютивна нейронна мережа, у якої відсутній останній прихований шар, що відповідає власне за класифікацію

зображень. Таким чином, результатом роботи нейронної мережі буде вектор ознак, які характеризують відповідне зображення, і, в той же час, цей вектор ознак буде використаний методом машинного навчання для класифікації у якості вхідної описової характеристики об'єкта.

Розглянемо основні класичні методи машинного навчання, що можуть бути використані в якості алгоритму на другому етапі – етапі побудови фінального класифікатора.

2.2.2.1 Лінійний класифікатор (Linear model)

Лінійні моделі – одні з найпоширеніших методів машинного навчання в цілому, які широко використовуються при вирішенні задач класифікації. Основна ідея будь якого лінійного класифікатора полягає у тому, щоб побудувати таку гіперплощину, яка б розділила вибірку на декілька підпросторів, кожен з яких відповідатиме певному значенню цільової змінної [14, ст.45].

Математично, найпростішу лінійну модель бінарної класифікації можна описати наступним чином:

$$y(x) = \text{sign} \left(\sum_{i=1}^n w_i * x_i \right) = \text{sign} \left(\vec{w}^T \vec{x} \right)$$

У даній моделі прогноз для об'єкту x будується шляхом визначення знаку лінійної комбінації вагових коефіцієнтів моделі \vec{w} та вхідного вектору ознак об'єкта \vec{x} . Відповідно, процес навчання такого класифікатора зводиться до знаходження вагових коефіцієнтів моделі.

Окремим, і найбільш відомим представником лінійних класифікаторів є логістична регресія.

Логістична регресія – окремий підвид лінійних моделей класифікації, що у випадку задачі бінарної класифікації має властивість прогнозувати не просто приналежність об’єкта до того чи іншого класу, а отримувати саме значення ймовірності того, що об’єкт, заданий даними описовими характеристиками належить до позитивного класу:

$$p_+ = P(y_i = 1 | x_i = \vec{x})$$

Об’єктами позитивного класу у задачах бінарної класифікації називаються ті об’єкти тренувальної вибірки, які мають значення цільової змінної – 1, негативного – 0. Розглянемо детальніше процес моделювання даної ймовірності та оцінки параметрів моделі логістичної регресії.

Розглянемо відношення:

$$B(\vec{x}) = \frac{p_+}{1 - p_+} \quad (2.1)$$

Тоді, так як логістична регресія – лінійна модель класифікації, отримаємо:

$$\log(B(x)) = \overline{w}^T \vec{x} \quad (2.2)$$

Застосовуючи співвідношення (2.1) та (2.2), можемо виразити ймовірність віднесення об’єкту \vec{x} до позитивного класу таким чином:

$$p_+ = \frac{B(\vec{x})}{1 + B(\vec{x})} = \frac{e^{\overline{w}^T \vec{x}}}{1 + e^{\overline{w}^T \vec{x}}} = \frac{1}{1 + e^{-\overline{w}^T \vec{x}}} = \text{sigmoid}(\overline{w}^T \vec{x}) \quad (2.3)$$

У формулі (2.3), $\text{sigmoid}(x)$ - сигмоїда. Таким чином, отримали ймовірність віднесення об’єкту \vec{x} до позитивного класу. У загальному випадку, ймовірність віднесення об’єкту \vec{x} до класу y_i зручно записувати наступним чином:

$$P(y = y_i | x_i = \vec{x}) = \frac{1}{1 + e^{y_i \vec{w}^T \vec{x}}}$$

Тепер розглянемо як власне відбувається процес навчання логістичної регресії. Запишемо ймовірність отримання вектору значень цільової функції на навчальній вибірці:

$$P(\vec{y} | X) = \prod_{i=1}^n P$$

Зазначимо, що записана ймовірність фактично залежить і від вектору вагових коефіцієнтів – параметрів моделі логістичної регресії, а отже остання рівність трансформується наступним чином:

$$P(\vec{y} | X) = \prod_{i=1}^n P(y = y_i \vee \vec{x} = \vec{x}_i, \vec{w}),$$

де \vec{y} – вектор значень цільової функції на навчальній вибірці;

X – матриця описових характеристик об'єктів початкової вибірки розмірністю $n \times m$, де n – кількість елементів у навчальній вибірці, m – кількість незалежних змінних – описових ознак об'єкта;

\vec{w} – вектор вагових коефіцієнтів логістичної регресії розмірністю $m \times 1$, який потрібно оцінити при навчанні моделі.

Використаємо метод максимальної правдоподібності для оцінки невідомих параметрів нашої навчальної вибірки:

$$\text{Log}(P(\vec{y} | X, \vec{w})) = \log \left(\prod_{i=1}^l P(y = y_i | \vec{x} = \vec{x}_i, \vec{w}) \right) =$$

$$\begin{aligned}
&= \sum_{i=1}^l \log(P(y = y_i | \vec{x} = \vec{x}_i, \vec{w})) = \\
&= \sum_{i=1}^l \log\left(\frac{1}{1 + e^{y_i \vec{w}^T \vec{x}}}\right) = - \sum_{i=1}^l \log(1 + e^{y_i \vec{w}^T \vec{x}})
\end{aligned}$$

Таким чином, задача максимізації логарифму функції правдоподібності зводиться до задачі мінімізації такого функціоналу:

$$\sum_{i=1}^n \log(1 + e^{y_i \vec{w}^T \vec{x}}) \quad (2.4)$$

Отриманий вираз (2.4) – логістична функція помилок, позначимо її наступним чином:

$$L_{\log}(X, \vec{y}, \vec{w}) = \sum_{i=1}^n \log(1 + e^{y_i \vec{w}^T \vec{x}})$$

Отже, ми звели процес навчання логістичної регресії до задачі мінімізації логістичної функції помилок:

$$\hat{w} = \operatorname{argmin}_{\vec{w}} (L_{\log}(X, \vec{y}, \vec{w})) \quad (2.5)$$

Окремим важливим аспектом навчання логістичної регресії, як і будь-якого іншого методу машинного навчання, є процес запобігання перенавчанню моделі – коли побудована модель фактично запам'ятала навчальну вибірку і втратила свою узагальнюючу здатність, тим самим унеможлививши використання такої моделі під час процесу прийняття рішень. Для запобігання перенавчанню моделей застосовується процес регуляризації, який для лінійних моделей буде мати такий вигляд:

$$\hat{w} = \operatorname{argmin}_{\vec{w}} (J(X, \vec{y}, \vec{w}) + \alpha \|\vec{w}\|) \quad (2.6)$$

де \hat{w} – оцінюваний вектор вагових коефіцієнтів лінійної моделі;

J – деяка функція помилок;

α – коефіцієнт регуляризації, що відповідає за степінь регуляризації та штрафу моделі.

Використовуючи співвідношення (2.5) та (2.6), напишемо остаточний запис задачі мінімізації, що розв’язується під час навчання моделі логістичної регресії:

$$\hat{w} = \operatorname{argmin}_{\vec{w}} (C * L_{\log}(X, \vec{y}, \vec{w}) + \|\vec{w}\|)$$

Даний вираз записаний таким чином, так як у випадку логістичної регресії прийнято у якості коефіцієнта регуляризації використовувати співвідношення $C = \frac{1}{\alpha}$. Таким чином, збільшуючи C , ми зменшуємо степінь регуляризації моделі.

Не менш важливим є поняття виду регуляризації, варіація якого задається нормою, що накладається на вектор параметрів моделі. У більшості випадків прийнято розглядати L_1 та L_2 регуляризації. Математично, вони виражаються так:

$$\|\vec{w}\|_{L_2} = \sum_{i=1}^n w_i^2, \quad \|\vec{w}\|_{L_1} = \sum_{i=1}^n |w_i|$$

Використання того чи іншого виду регуляризації сильно залежить від поставленої задачі та типу вхідних даних. Наприклад, коли вектор вхідних ознак об’єкту навчальної вибірки є багатовимірним, то буває корисно використати L_1 -регуляризацію, так як даний вид регуляризації має властивість обнуляти коефіцієнти при деяких ознаках об’єкту, тим самим здійснюючи автоматичний відбір ознак, потрібних для прийняття рішення.

Лінійні моделі класифікації, незважаючи на всю їх простоту, мають один суттєвий недолік: вони працюють у припущенні, що вибірка є лінійно роздільною. Наступний метод, який ми розглянемо, не має таких недоліків.

2.2.2.2 Метод опорних векторів (Support Vector Machines)

Метод опорних векторів – лінійна модель бінарної класифікації, головна ідея якої полягає у побудові гіперплощини у n -вимірному просторі, яка б найкраще розділяла задані класи об'єктів. Ця гіперплощина підбирається таким чином, щоб максимізувати відстань між нею та найближчими з об'єктів кожного класу. Дані об'єкти, які є найближчими до роздільної гіперплощини отримали назву – опорні вектори – звідси і походить назва даного методу [14, ст.92]. Розглянемо більш детально принцип роботи методу опорних векторів.

Нехай задана тренувальна вибірка: X – матриця ознак об'єктів тренувальної вибірки, \vec{y} – вектор значень цільової функції на об'єктах тренувальної вибірки. Ключовим фактором для роботи даного методу є те, що \vec{y} приймає значення «-1» для об'єктів негативного класу та «+1» для об'єктів позитивного класу.

Для того, щоб максимізувати відстань між гіперплощиною та опорними векторами, вводиться таке поняття як проміжок (margin) між об'єктами класів, який визначається так:

$$M(x_i) = y_i (\vec{w}^T \vec{x} - \vec{w}_0), \quad (2.7)$$

де x_i – деякий об'єкт навчальної вибірки;

y_i – відповідне значення цільової змінної;

\vec{w}^T – вектор вагових коефіцієнтів моделі;

\vec{x} – вектор ознак для об'єкта x_i ;

\vec{w}_0 – деякий скалярний поріг, що визначає роздільну гіперплощину.

Розглянемо \vec{w} – вектор нормалі, проведений до роздільної гіперплощини. Знайдемо проекцію вектору, кінцями якого будуть опорні вектори, на вектор \vec{w} . Довжина цієї проекції і буде показувати ширину роздільної полоси, яку алгоритму потрібно максимізувати під час навчання:

$$\begin{aligned}
\langle (x_+ - x_-), \frac{\vec{w}}{\|\vec{w}\|} \rangle &= \frac{1}{\|\vec{w}\|} * (\langle x_+, \vec{w} \rangle - \langle x_-, \vec{w} \rangle) = \\
&= \frac{2}{\|\vec{w}\|} \rightarrow \max \rightarrow \|\vec{w}\| \rightarrow \min \leftrightarrow \frac{(\vec{w}^T \vec{w})}{2} \rightarrow \min
\end{aligned} \tag{2.8}$$

де x_+ – об'єкт позитивного класу;

x_- – об'єкт негативного класу;

\vec{w} – вектор нормалі.

Ілюстрацію алгоритму навчання методу опорних векторів можна побачити на рисунку 2.9.

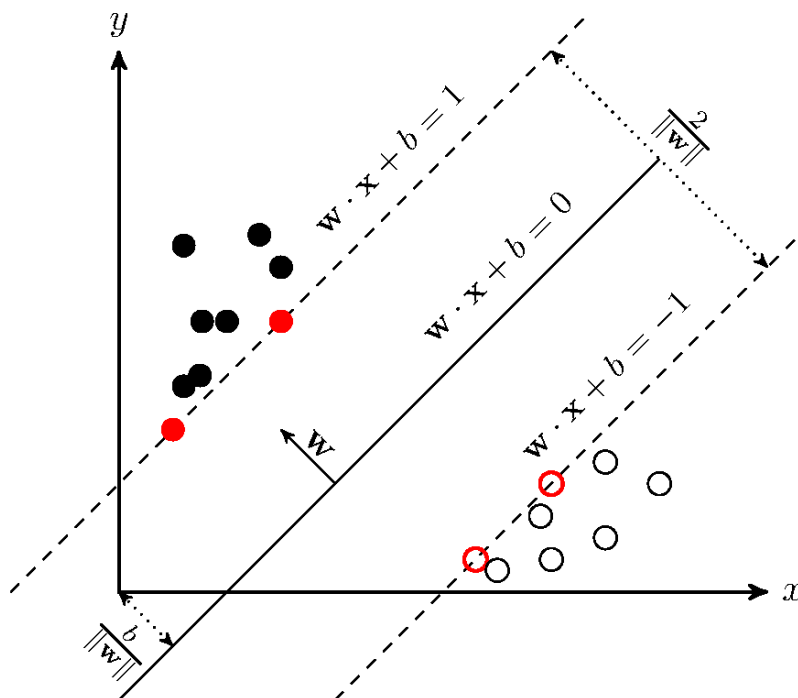


Рисунок 2.9 – Принцип роботи алгоритму побудови максимальної роздільної полоси або [15]

Розглянемо більш детальніше формалізацію відступу. Як видно із співвідношення (2.7), відступ буде від'ємним тоді і тільки тоді, коли алгоритм допускає помилку в класифікації. Якщо значення відступу знаходиться на проміжку від 0 до 1, то класифікований об'єкт потрапляє на роздільну гіперплощину, що говорить про те, що модель є невпевненою у класифікації даного об'єкта. Якщо ж об'єкт класифіковано правильно, то значення відступу буде більше 1.

Об'єднуючи це з формулою (2.8), сформулюємо задачу оптимізації, яка розв'язується при навчанні методу опорних векторів:

$$\begin{cases} \frac{(\vec{w}^T \vec{w})}{2} \rightarrow \min \\ \vec{y} (\vec{w}^T \vec{x} - \vec{w}_0) \geq 1 \end{cases} \quad (2.9)$$

Задача (2.9) розв'язується аналітично: за теоремою Куна-Такера. Вона еквівалентна двоїстій задачі пошуку сідлової точки функції Лагранжа:

$$\begin{cases} L(\vec{w}; \vec{w}_0; \vec{\lambda}) = \frac{(\vec{w}^T \vec{w})}{2} - \sum_{i=1}^n \lambda_i (y_i (\langle \vec{w}, \vec{x}_i \rangle - w_0) - 1) \rightarrow \min_{\vec{w}, \vec{w}_0} \max_{\vec{\lambda}}; \\ \lambda_i \geq 0, i = 1 \dots n \\ \begin{cases} \lambda_i = 0 \\ \langle \vec{w}, \vec{x}_i \rangle - w_0 = y_i \end{cases}, i = 1 \dots l \end{cases},$$

де $\vec{\lambda} = (\lambda_1, \dots, \lambda_l)$ – вектор двоїстих змінних.

Наведений алгоритм застосовується у випадку лінійної роздільності об'єктів двох класів.

У випадку лінійної нероздільності класів, метод опорних векторів спочатку проектує дані у простір вищої розмірності, допоки вибірка не стане лінійно роздільною, а потім застосовує алгоритм, наведений вище. У загальному випадку, етап проєкції даних у простір більш високої розмірності є дуже трудомістким процесом, що потребує великих обчислювальних можливостей, адже при цьому будуть розраховуватись різноманітні комбінації нелінійних змінних, що формуються на основі ознак об'єкта. У свою чергу, метод опорних векторів використовує так званий «ядровий трюк» (kernel trick), за допомогою якого відбувається проєкція даних у простір більш високої розмірності без фактичного розрахунку розширених ознак об'єкта. Ядро – це матриця K , яка складається із скалярних добутків пар векторів \vec{x}_i та \vec{x}_j :

$$K(x_i, x_j) = \langle \psi(x_i), \psi(x_j) \rangle$$

Найчастіше використовують наступні ядрові функції:

а) лінійне ядро:

$$K(x_i, x_j) = x_i^T x_j$$

б) поліноміальне ядро зі степеню p :

$$K(x_i, x_j) = (1 + x_i^T x_j)^p$$

в) гаусівське ядро з радіально-базисною функцією або RBF (Radial Basis Function):

$$K(x_i, x_j) = \exp\left(\gamma \|x_i - x_j\|^2\right)$$

г) сигмоїдне ядро:

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + \beta_0)$$

Говорячи про властивості описаних ядрових функцій, можна виділити такі:

- а) при використанні поліноміального ядра обчислюються усі можливі поліноміальні комбінації вихідних ознак об'єкта до певної степені;
- б) при використанні гаусівського ядра обчислюються усі можливі поліноміальні комбінації усіх степенів щодо вихідних ознак об'єкта.

Таким чином, метод опорних векторів працює в два етапи: спочатку проектує вихідні ознаки об'єкта у простір більш високої розмірності, де дана вибірка буде лінійно роздільна, а потім працює як лінійний класифікатор, будуючи роздільну гіперплощину шляхом максимізації відступу.

2.2.2.3 Древа прийняття рішень (Decision Trees)

Древа прийняття рішень – вид алгоритмів машинного навчання з учителем, який належать до сім'ї методів машинного навчання, які засновані на правилах (rule-based algorithms). Даний вид методів характерний тим, що отримує знання з даних шляхом побудови емпіричних правил, які легко інтерпретувати, виходячи із вхідних даних. Древа прийняття рішень також можна адаптувати як для задач класифікації, так і для задач регресії [14, ст.70].

Розглянемо основні структурні елементи дерева прийняття рішень:

- а) вузол (node) – основний архітектурний елемент дерева, який містить предикат, що розбиває множину елементів вузла на дві підмножини;
- б) лист – вузол дерева, що не має дочірніх елементів, який у задачах класифікації буде містити класифіковані елементи вихідної множини об'єктів.

Говорячи про принципи побудови дерева, розглянемо два основних алгоритми, які використовуються на практиці: C4.5 та CART.

Розглянемо принцип роботи алгоритму C4.5. Даний метод побудови дерева є жадібним алгоритмом і працює на основі зменшення ентропії множини елементів.

Нагадаємо, що ентропія множини X розраховується за формулою Шеннона:

$$H(X) = - \sum_{i=1}^n p_i * \log_2 p_i,$$

де p_i – ймовірність отримання елемента того чи іншого класу;

n – кількість класів.

Зменшення ентропії фактично означає те, що множина стає більш однорідною, а це означає, що модель правильно класифікує об'єкти навчальної вибірки. На кожній ітерації побудови дерева, алгоритм будує такий предикат, щоб ентропія множини після розбиття була найменшою. Алгоритм закінчує свою роботу, коли зменшення ентропії є несуттєвим і не несе великої предикативної сили.

Серед недоліків даного алгоритму можна виділити те, що побудоване дерево виходить дуже «глибоким», а це може привести до перенавчання та побудови дуже складних предикатів, які будуть відображати лише ті взаємозв'язки, що є в навчальній вибірці. Дану проблему вирішує більш модифікований алгоритм – CART.

Недоліками дерев прийняття рішень є те, що вони здатні швидко перенавчатись і є менш робастними.

Дані недоліки вирішуються завдяки побудові ансамблів дерев рішень. Основними методами ансамблювання моделей є *boosting* та *bootstrap aggregating* [14, ст.83].

Перший метод полягає у тому, щоб ітеративно будувати дерева прийняття рішень, вибираючи у тренувальну вибірку ті елементи вихідної навчальної вибірки, на яких модель найбільше помилялась у ході попередніх ітерацій.

Другий метод полягає у формуванні з вихідної навчальної вибірки n незалежних вибірок з повтореннями. На кожній із цих вибірок незалежно будується дерево прийняття рішень. Далі, отримані класифікатори об'єднуються у фінальний класифікатор шляхом голосування.

У наступному розділі проведемо порівняльну характеристику описаних методів.

2.3 Порівняльний аналіз методів класифікації зображень

Розглянемо порівняльну характеристику наведених методів (таблиця 2.1).

Таблиця 2.1 – Порівняльна характеристика методів класифікації зображень

Назва критерію	Конволютивна нейронна мережа	Трансферне навчання + метод опорних векторів	Трансферне навчання + логістична регресія	Трансферне навчання + методи, засновані на деревах прийняття рішень
Додаткова генерація ознак	-	+	+	+
Висока швидкість навчання	-	-	+	-
Можливість навчання на даних високої розмірності	+	+	+	-
Висока ймовірність перенавчання	+	-	-	+
Точність прогнозування	+	+	-	+

Джерело: розроблено автором

Як бачимо з порівняльної таблиці, кожен із представлених методів має свої недоліки та переваги. Найбільш важливими аспектами при оцінці даних моделей є потреба генерування додаткових ознак як вхідних ознак для класифікаційного алгоритму, а також підтримання високої точності прогнозування. Зважаючи на дані критерії, в якості основного алгоритму, який буде використовуватись в роботі, було обрано конволютивні нейронні мережі, які, водночас, будуть скореговані для прогнозування конкретного виду захворювання за вхідним зображенням.

2.4 Критерії оцінки якості роботи моделі класифікації зображень

Розглянемо основні метрики, які використовуються для оцінки моделей класифікації.

Матриця помилок (confusion matrix) – це таблиця, що часто використовується для ілюстрації якості роботи класифікатора (рис.2.10).

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Рисунок 2.10 – Матриця помилок класифікаційної моделі або [16]

Дана матриця фактично є повним описом результатів класифікації, що дозволяє побачити кількість правильно та неправильно класифікованих об'єктів. Досить важливою властивістю такого методу оцінки моделей є те, що багато інших метрик для класифікації виводяться за її допомогою.

Ассигасу метрика є найбільш популярною при вирішенні задач класифікації. Це обумовлено простотою розрахунків та інтерпретації результатів роботи. Фактично, ця метрика показує кількість правильних відповідей класифікаційної моделі на певному наборі даних. Серед недоліків слід виділити погану якість роботи на незбалансованих наборах даних.

Математично, дану метрику можна описати так:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision метрика показує те, як точно модель класифікує об'єкти позитивного класу. Фактично, точність показує величину помилки першого роду, яку робить класифікаційна модель. Наведемо її математичну формулу:

$$precision = \frac{TP}{TP + FP}$$

Повнота (recall, true positive rate) – цей показник ілюструє те, як модель виділяє об'єкти позитивного класу. Низькі значення повноти фактично означають високу помилку другого роду: класифікаційна модель не виділяє об'єкти позитивного класу. Високі значення навпаки, можуть означати те, що у моделі висока помилка першого роду. Математична формула:

$$recall = \frac{TP}{TP + FN}$$

Говорячи про точність та повноту, ми розуміємо, що збільшення одного показника часто веде до низьких результатів іншого. F-метрика (F_β score) дозволяє при оцінці моделей одночасно враховувати як точність, так і повноту. F-метрика – зважене гармонійне середнє між точністю та повнотою:

$$F_\beta = (1 + \beta^2) * \frac{precision * recall}{\beta^2 * precision + recall}$$

При $\beta > 1$, метрика є більш чутливою до змін повноти, а при $\beta < 1$ – точності.

При $\beta = 1$, точність і повнота є рівнозначними при розрахуванні значення метрики.

ROC (receiver operating characteristic) – дана метрика фактично є кривою, що будується на координатній площині, де за вісь ординат відповідає повнота, а за вісь абсцис – відношення кількості помилкових спрацювань моделі на позитивний клас до кількості об'єктів негативного класу у вибірці. Побудувавши дану криву в

одиночному квадрати, розраховують площу під кривою i , фактично, отримують власне значення даної метрики. Цей показник досить часто використовується в задачах класифікації, але його не варто використовувати у випадках нерівномірного розподілу класів у вибірці.

Поставлена у роботі задача диктує і вибір коректної метрики для оцінки моделей класифікації, адже при проведенні медичної діагностики дуже важливо мінімізувати помилку другого роду – тобто кількість випадків, у яких модель «пропускатиме» позитивний клас, некоректно ідентифікуючи можливе захворювання. Водночас, можливе збільшення помилки першого роду, і, як наслідок, підвищення кількості випадків «хибного спрацювання» моделі є не так важливим, адже набагато гірше не помітити можливий прояв хвороби, аніж помилково здоровому пацієнту вказати на те, що він має деякі прояви того чи іншого захворювання.

Таким чином, можна зробити висновок, що оптимальним критерієм оцінки побудованих моделей буде F-метрика.

2.5 Метод проведення медичного діагностування захворювань

Відповідно до теми роботи, необхідно реалізувати декілька алгоритмів класифікації зображень. Оскільки сервіс, який потрібно спроектувати є мультизадачним: ми проводимо медичне діагностування для декількох видів хвороб, то задача класифікації буде трансформуватись у задачі як бінарної, так і мультикласової класифікації. Дана специфікація буде визначати і архітектуру фінальних конволютивних нейронних мереж, що будуть використовуватись у якості класифікаторів для кожного із розглянутих захворювань.

2.6 Висновки до розділу 2

У даному розділі було наведено основні види методів машинного навчання, а також сформульовано принцип роботи методів машинного навчання з учителем.

Розглянуто задачу класифікації як підвид методів навчання з учителем, а також проведено опис та порівняльну характеристику методів її розв'язання. Окремо було наведено основні метрики оцінювання класифікаційних моделей, а також аргументовано вибір методу та метрики для вирішення задачі класифікації.

РОЗДІЛ 3 АНАЛІЗ АРХІТЕКТУРИ ТА ІНТЕРФЕЙСУ ПРОГРАМНОГО ПРОДУКТУ

3.1 Обґрунтування вибору платформи та мови програмування

Розробку сервісу та алгоритмів класифікації було вирішено здійснювати за допомогою мови програмування Python із використанням фреймворку Django. Python – це високорівнева, об’єктно-орієнтована мова програмування, без жорсткої типізації, яка має широкий функціонал з «коробки», що дає змогу реалізувати обширну частину програмного продукту без використання додаткових залежностей. Також дана мова програмування має ще одну визначальну у порівнянні з іншими мовами програмування особливість: наявність великої кількості пакетів для обробки та візуалізації масивів даних, розробки алгоритмів машинного навчання, зокрема глибинного навчання, підтримки платформ для обробки значних об’ємів інформації, таких як Spark чи Hadoop.

Розглядаючи функціонал, необхідний для вирішення поставленої задачі, потрібно зазначити і наявність таких пакетів для роботи з глибинним навчанням як Keras та Pytorch, а також спеціалізованого пакету для обробки зображень – scikit-image. Для використання методів класичного машинного навчання існує модуль scikit-learn, у якому реалізовані абстракції усіх основних алгоритмів. У роботі було використано Keras, що працює із використанням TensorFlow – спеціалізованого пакету для проведення математичних операцій з багатовимірними тензорами [17].

Неодмінним атрибутом при розробці алгоритмів машинного навчання та роботі з даними також є Jupyter Notebook – інтерактивна платформа, що дозволяє швидко проводити обчислення та візуалізувати отримані результати.

Так як програмний продукт передбачає збереження отриманих результатів до бази даних, то не менш важливим фактором при виборі мови програмування є можливість швидкої та зручної інтеграції з базами даних. Python є мовою програмування, що зокрема використовується для написання бекенд (backend)

частини багатьох програмних продуктів, а отже має можливість роботи із будь-якими найпопулярнішими видами баз даних, як лінійно-орієнтованих (PostgreSQL, MySQL, SQLite тощо), так і вертикально-орієнтованих (Vertica).

Слід зазначити, що так як програмний продукт є досить складним і багатофункціональним, то для контролю коректної роботи кожного елемента архітектури сервісу – від обробки та власне класифікації зображень до збереження результатів роботи програми у базу даних – було реалізовано систему логування із використанням засобів Python.

Тож, ураховуючи усі ці особливості, було вирішено використовувати саме дану мову програмування.

Розглядаючи фреймворки, необхідні для якісної реалізації сервісу, що буде використовувати побудовані алгоритми машинного навчання для вирішення практичної задачі, то Python надає такі можливі варіанти: Django, Flask, CherryPy та інші. Найбільш популярними є перші два – Django та Flask.

Flask – фреймворк, що зазвичай використовується для реалізації так званих Restful API – розподілених програмних систем, що працюють за принципом клієнт-серверних додатків. Даний фреймворк є дуже корисним при розробці продуктів, які повинні отримувати деякі вхідні дані, розрахувати на їх основі відповідні показники та віддавати результат користувачу. Водночас, Flask є досить незручним, коли потрібно, окрім бекенд частини, займатись розробкою і підтримкою фронтенд (frontend) частини продукту, а також налагоджувати інтеграцію з базами даних. Зокрема, у якості прикладу можна привести і той факт, що Flask не має вбудованої системи відпрацювання міграцій, а отже кожна зміна у структурі бази даних, що використовується у проекті має бути реалізована та протестована із написанням власного коду, що створює додаткові незручності при збільшенні об'єму процесів і всього проекту.

Django – фреймворк, що застосовується для повноцінної побудови сервісів, причому як їх бекенд так і фронтенд частини. Серед основних переваг цього фреймворку є надання вбудованої лінійно-орієнтованої бази даних – SQLite, а також можливість інтеграції з усіма популярними базами даних. Ще однією

визначальною особливістю Django є використання ORM – концепції програмування та адміністрування базами даних, що дозволяє перенести та відтворити основні властивості об'єктно-орієнтованого програмування на роботу з базами даних, створюючи таким чином об'єктну базу даних. Дана концепція є дуже зручною та ефективною при налагодженні взаємодії роботи основної архітектури програмного продукту з базами даних. Також слід зазначити, що даний фреймворк має багато вбудованих алгоритмів для роботи з мережевими запитами, що робить його використання ще більш зручнішим та ефективнішим.

Отже, ураховуючи усі особливості розглянутих засобів, у роботі було вирішено використовувати Django, як найбільш відповідний варіант.

3.2 Аналіз вхідних даних та керівництво користувача

Побудований сервіс буде використовуватись для проведення медичної діагностики на такі захворювання як діабетична ретинопатія, крововилив у мозок та пневмонія. Відповідно до цього, вхідні дані будуть представлятись у вигляді трьохканальних (RGB) кольорових зображень захворюваної області тіла розмірностями 224 на 224, 224 на 224 та 128 на 128 відповідно. Проведемо базовий експлоративний аналіз кожного виду вхідних даних.

Перший аналіз – це задача діагностування діабетичної ретинопатії за зображенням сітчатки ока. Дана задача є задачею мультикласової класифікації, а отже включає в себе декілька класів вхідних зображень:

- а) «0» позначає клас зображень, в яких показані абсолютно здорові зображення сітчатки, без наявності ознак захворюваності;
- б) «1» позначає клас зображень, в яких показані зображення сітчатки з легкою формою ретинопатії;
- в) «2» позначає клас зображень, в яких показані зображення сітчатки з помірною формою ретинопатії;

г) «3» позначає клас зображень, в яких показані зображення сітчатки з важкою формою ретинопатії;

г) «4» позначає клас зображень, в яких показані зображення сітчатки з профілеративною формою ретинопатії;

Як бачимо із рис. 3.1., дана задача має незбалансований розподіл цільової змінної.

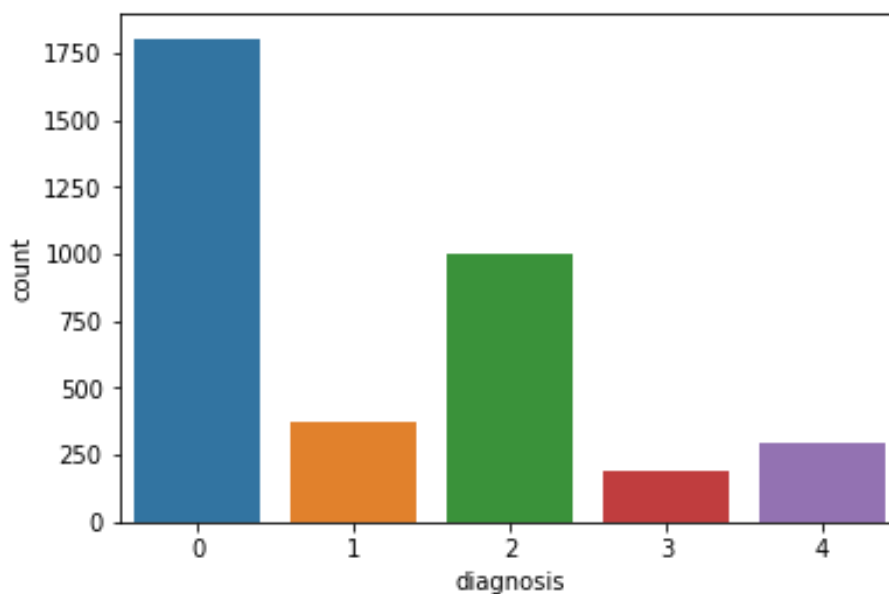


Рисунок 3.1 – Розподіл кількості класів зображень у задачі діагностування діабетичної ретинопатії

Джерело: розроблено автором

На вхід алгоритму класифікації подаються кольорове зображення сітчатки ока, як показано на рисунку 3.2.



Рисунок 3.2 – Приклад вхідного зображення сітчатки ока у задачі діагностування діабетичної ретинопатії або [18]

Другий аналіз – задача діагностування пневмонії за зображенням флюорографії легень (див. рис. 3.3).



Рисунок 3.3 – Приклад вхідного зображення флюорографії легень для діагностики пневмонії або [19]

Дана задача є задачею бінарної класифікації, тобто цільова змінна буде набувати лише двох значень: наявність та відсутність пневмонії відповідно. Для тренування були використані 5863 зображення, що були зібрані протягом 5 років досліджень. Кожне із вхідних зображень було очищено від додаткового шуму та провалідоване двома незалежними експертами.

Третій аналіз – задача діагностування крововиливу в мозок за зображенням магнітно-резонансної томографії мозку. Дана проблема є також задачею мультикласової класифікації. Вхідні зображення є конвертованими зображеннями із типу DICOM в PNG. Загальний вигляд даних зображень показаний на рисунку 3.4.

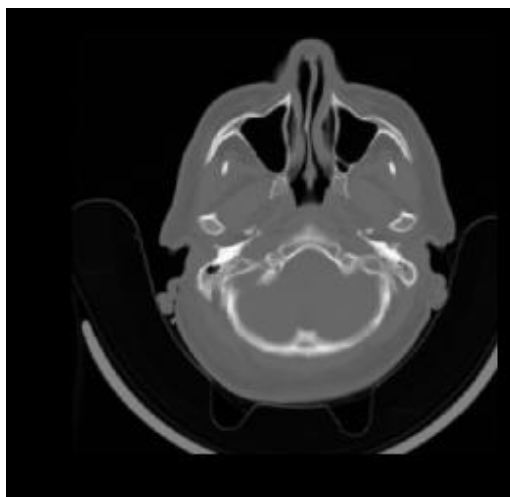


Рисунок 3.4 – Приклад зображення магнітно-резонансної томографії мозку у наборі даних або [20]

Розглянемо мітки класів, які використовуються в даній задачі:

- а) «1» позначає епідуральний крововилив;
- б) «2» позначає внутрішньопаренхімальний крововилив;
- в) «3» позначає внутрішньошлуночковий крововилив;
- г) «4» позначає субдуральний крововилив;
- г) «0» позначає зображення, що немає ознак захворюваності.

Також слід зазначити, що вибірка в цілому є сильно неоднорідною: деякі мітки класів вкрай рідко зустрічаються у вибірці і фактично є аномальними (див. рис. 3.5).

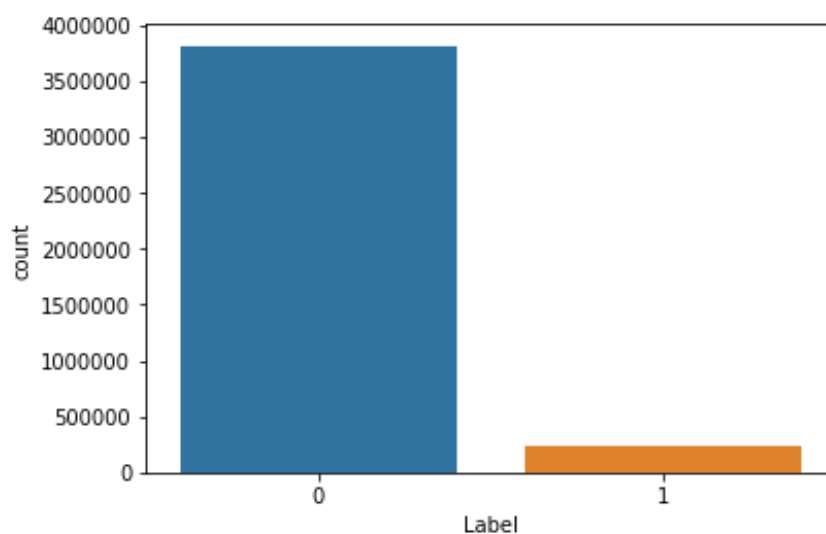


Рисунок 3.5 – Розподіл цільової змінної у розрізі наявності та відсутності захворювання відповідно

Джерело: розроблено автором

Розроблений сервіс включатиме усі необхідні аспекти та властивості для того, щоб зробити його найбільш привабливим та зручним для користувача. Таким чином, основний процес взаємодії користувача та сервісу буде побудований із використанням рішень фреймворку Django. Основними діями, що потрібно буде виконати користувачу є такі:

- а) перехід на сторінку для проведення діагностики;
- б) вибір проблеми класифікації та завантаження відповідного зображення;
- в) реєстрація на формі для реєстрації.

Далі, після відправки зображення на опрацювання сервісом, користувачеві буде показана сторінка виведення результатів із відповідними коментарями.

3.3 Аналіз архітектури програмного продукту

Архітектура побудованого сервісу складається з двох великих архітектурних елементів:

- а) бекенд частина, що містить програмну реалізацію попередньої обробки вхідного зображення, його класифікацію відповідно до обраної проблеми, а також збереження результатів роботи до бази даних;
- б) фронтенд частина, що відповідає за користувацький інтерфейс: можливість користувача пересуватись по сторінкам сервісу, переходити на сторінку реєстрації та здійснювати авторизацію, здійснювати завантаження необхідного зображення.

Розглянемо кожен із даних структурних елементів більш детально.

Основними елементами фронтенд-архітектури сервісу є такі процеси:

- а) реєстрація та авторизація користувача;
- б) завантаження зображення та основних метаданих для здійснення класифікації;
- в) вивід отриманих результатів.

Опишемо кожен із них у термінах об'єктно-орієнтованого програмування.

Реєстрація та авторизація користувача реалізована за допомогою абстракції-моделі General, що наслідується від базового класу `django.model.Model` та пов'язана з базовим класом для реєстрації користувачів у Django – `djnago.contrib.auth.User` – типом зв'язку один до одного. Фактично, даний клас є розширенням базової абстракції користувача і має такі атрибути:

- а) `user` – об'єкт класу `djnago.contrib.auth.User`, що має основні поля та методи, необхідні для реєстрації та авторизації користувачів;
- б) `age` – поле типу `integer`, що використовується для зберігання інформації про вік користувача;
- в) `country` – поле типу `string`, що містить інформацію про країну проживання користувача.

Методи даного об'єкта будуть описувати весь необхідний функціонал для роботи з даним класом.

Завантаження зображення та метаданих для подальшої класифікації відбувається шляхом використання класу `Detection`, що є дочірнім до класу `django.model.Model` та має наступні поля-атрибути:

- а) `name` – поле типу `string`, що зберігатиме ім'я користувача;
- б) `email` – поле типу `django.emailField` яке містить електронну поштову адресу користувача;
- в) `subject` – поле типу `string`, що містить тип класифікаційної проблеми, яка вирішується;
- г) `img_to_detect` – поле типу `django.imageField`, яке використовується для завантаження відповідного зображення.

Слід також зазначити, що поля `name` та `email` є унікальними ідентифікаторами користувача і використовуються у якості первинних ключів у відповідних таблицях бази даних для ефективного збереження та пошуку інформації. Також дані атрибути використовуються для ідентифікації факту реєстрації користувача у системі. У разі, якщо користувач не є зареєстрованим у системі, то після завантаження зображення сервіс розрахує результат класифікації та

збереже його до бази даних, але користувач буде переміщений на сторінку реєстрації, без можливості отримання результатів роботи сервісу. Лише після проведення реєстрації, користувач зможе отримати відповідні результати класифікації.

Вивід отриманих результатів роботи сервісу відбувається із використанням класу ResultsDetectionModel, що наслідується від `django.db.models.Model` та має тільки одне поле типу `string` – `result`, що виводиться користувачеві на сторінку візуалізації результатів. Отже, описану архітектуру можна зобразити у вигляді наступних діаграм класів (рис. 3.6 – 3.8).

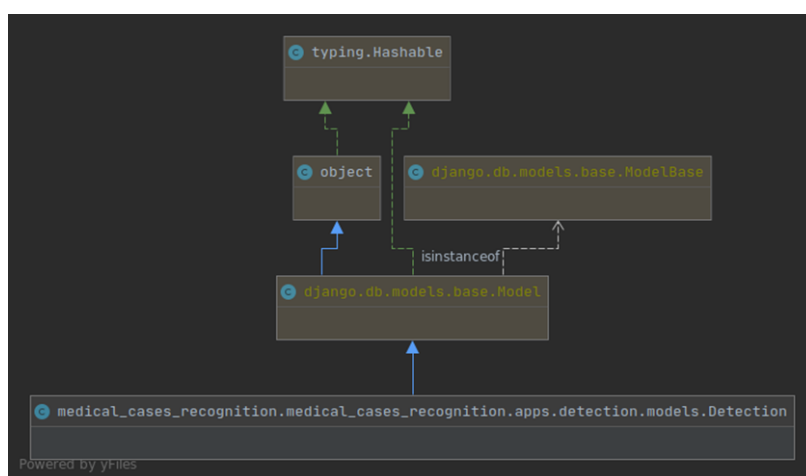


Рисунок 3.6 – Діаграма класів для класу Detection

Джерело: розроблено автором

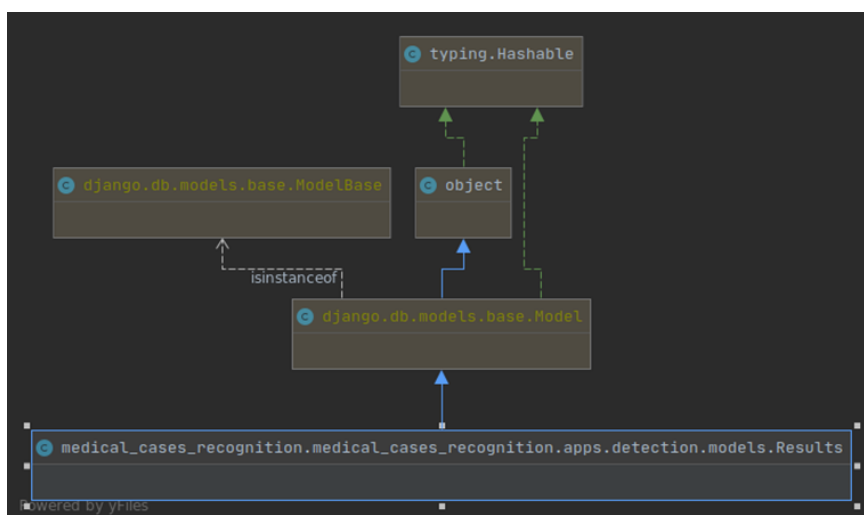


Рисунок 3.7 – Діаграма класів для класу Results

Джерело: розроблено автором

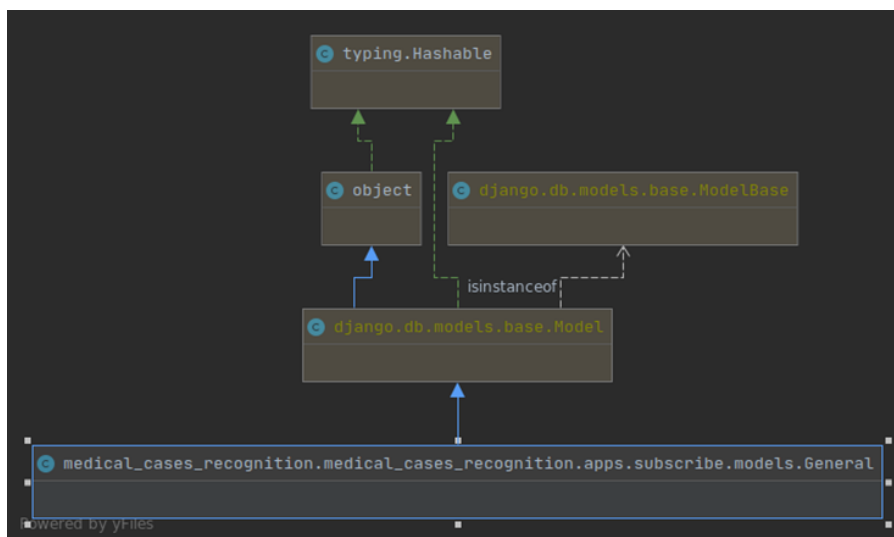


Рисунок 3.8 – Діаграма класів для класу General

Джерело: розроблено автором

Бекенд-частина, у свою чергу, включає в себе процес класифікації та інтеграції з базою даних.

Процес класифікації зображення розбитий на два етапи:

- а) проведення попередньої обробки зображення: конвертація значень масиву пікселів до цілих чисел, трансформація зображення до потрібної розмірності;
- б) отримання розрахованих ймовірностей за допомогою нейронної мережі.

Інтеграція з базою даних значною мірою відбувається за рахунок вбудованих процесів Django. Слід зауважити, що так як даний фреймворк використовує ORM архітектуру для взаємодії з базою даних, то кожен користувацький клас є водночас і моделлю бази даних, що повністю визначає архітектуру та властивості відповідних таблиць та алгоритм проведення міграцій при внесенні змін у сховище даних.

Для створення максимально зручного та гнучкого у використанні функціоналу, було створено базу даних засобами стандартної бази даних SQLite, діаграму якої наведено нижче (див. рис. 3.9).

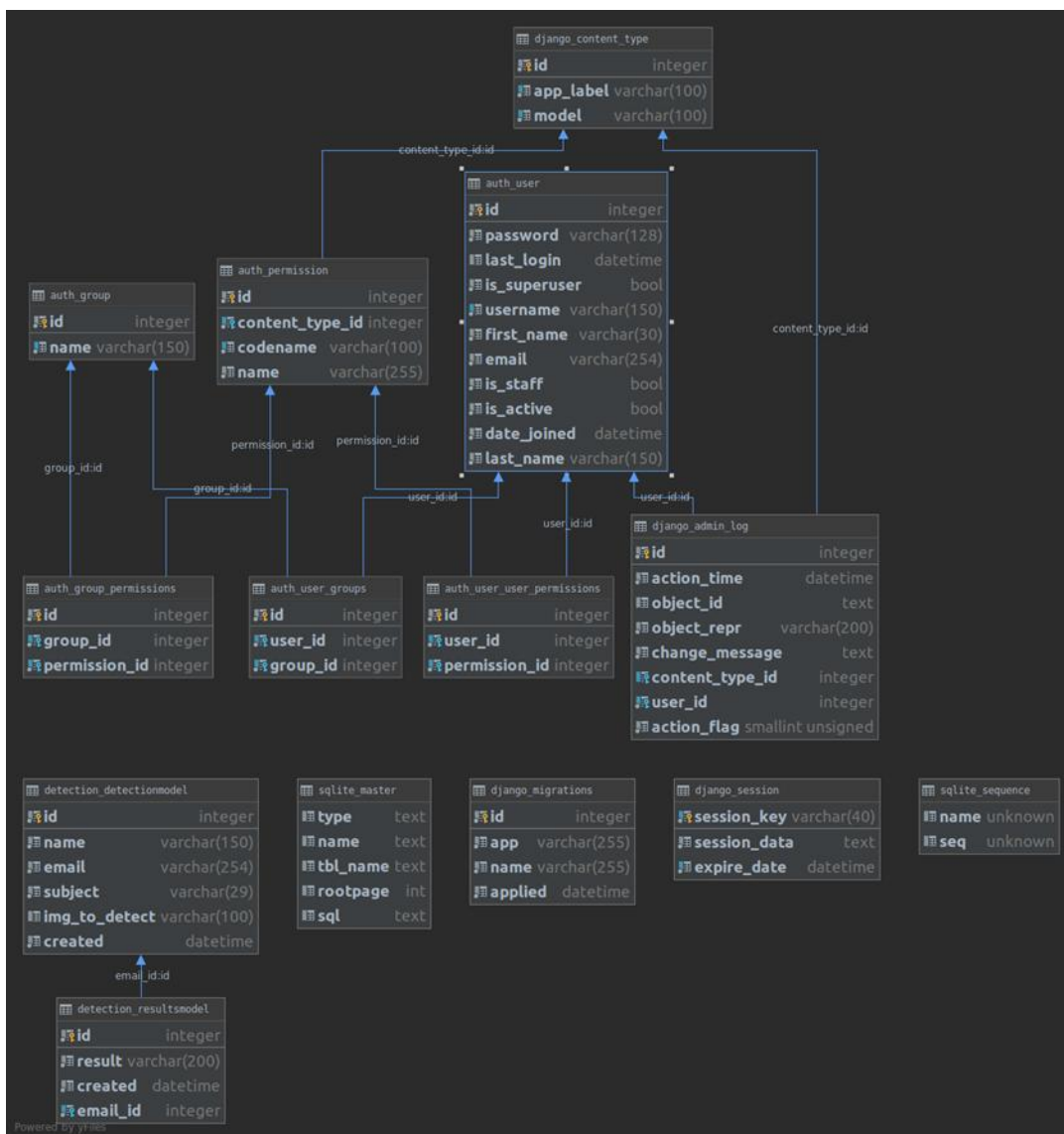


Рисунок 3.9 – Діаграма бази даних

Джерело: розроблено автором

Як бачимо, побудована база даних повністю відтворює об'єктну структуру описаної вище архітектури. Окрім цього, шляхом використання архітектури ORM, створено структуру допоміжних таблиць, які використовуються для проведення транзакцій, міграції та інших критично важливих операцій взаємодії з базою даних, що можуть суттєво вплинути на продуктивність її роботи. Таким чином, побудована архітектура сховища бази даних є відносно стійкою до великого навантаження на систему і не спричинить суттєвих збоїв у роботі під час перевантаження сервісу запитами на обробку інформації.

3.4 Опис інтерфейсу програмного продукту

Розробка інтерфейсу програмного продукту значною мірою включала в себе розв'язання багатьох класичних задач проектування його зовнішньої архітектури. Дана задача вирішувалась шляхом використання спеціальних методів, реалізованих Django, а також із застосуванням спеціальних пакетів для роботи з CSS та Bootstrap.

Інтерфейс складається з таких основних архітектурних елементів:

- а) домашня сторінка сервісу;
- б) реєстрація та авторизація користувача;
- в) вибір типу проведення медичної діагностики;
- г) виведення результатів.

Домашня сторінка (рис.3.9) розробленого програмного продукту складається із декількох точок входу на інші сторінки сервісу, а також на реєстрацію користувача.

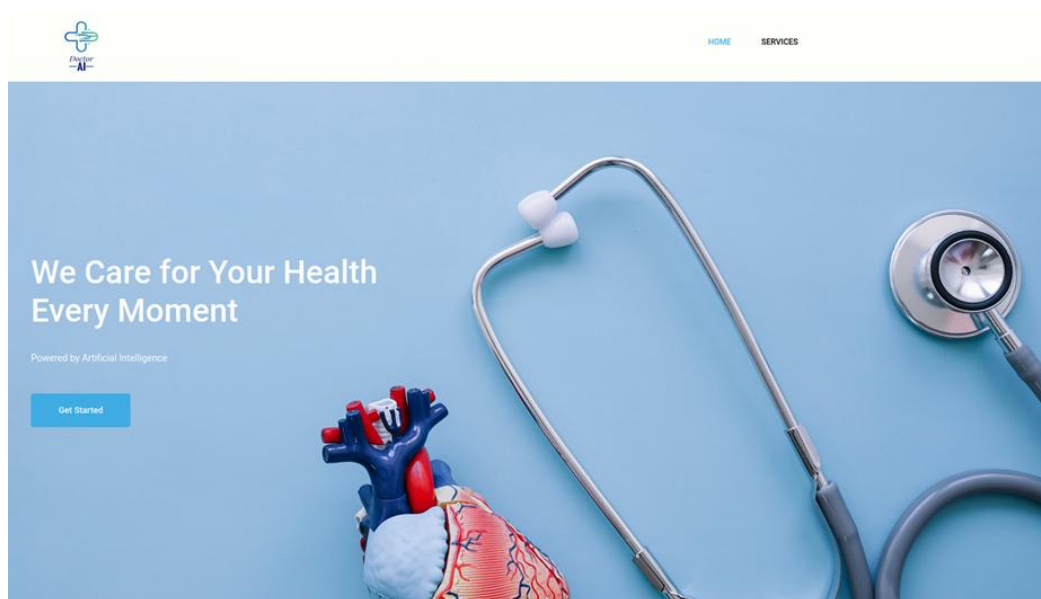
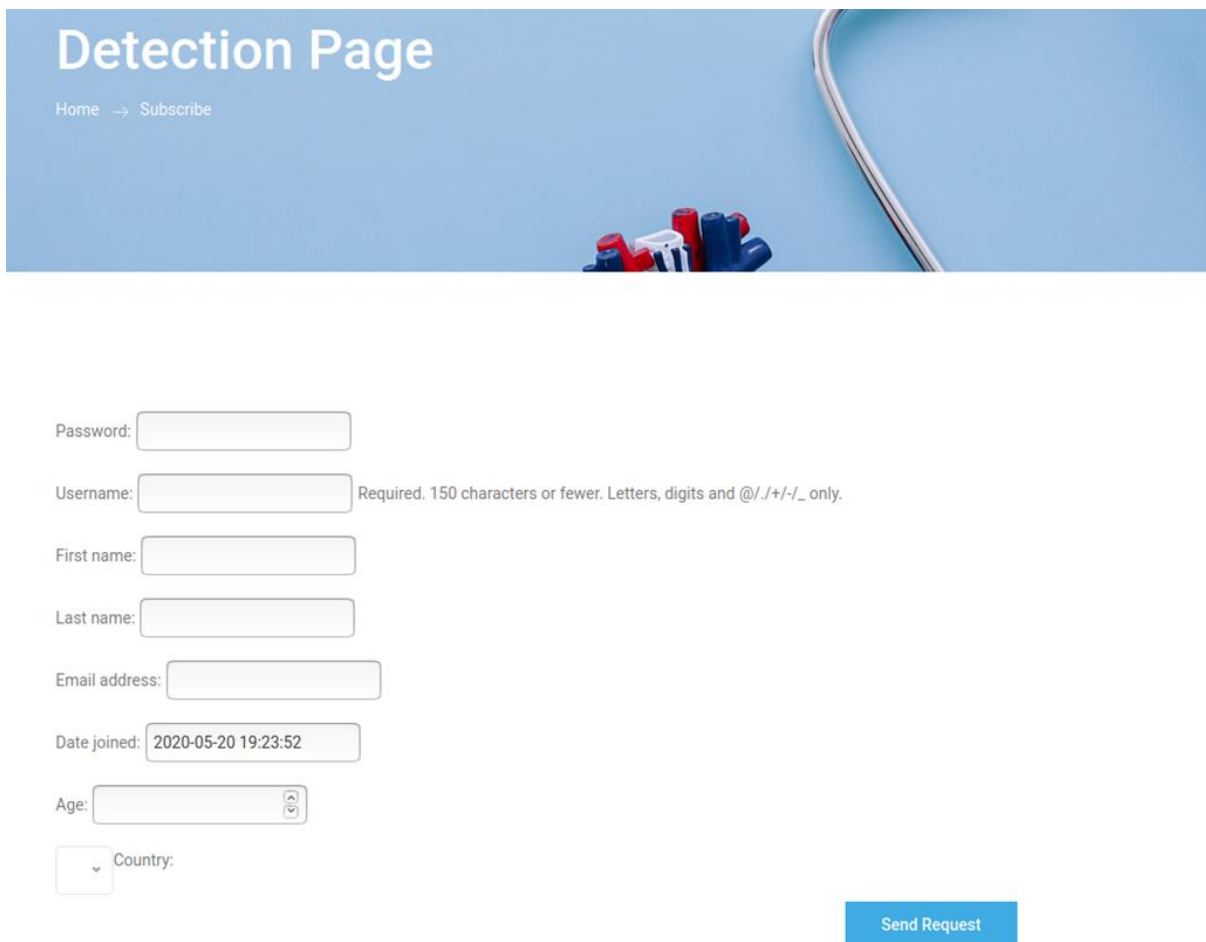


Рисунок 3.9 – Вигляд домашньої сторінки розробленого сервісу

Джерело: розроблено автором

Сторінка реєстрації користувача складається із основних полів, необхідних для отримання усієї необхідної базової інформації про користувача: ім'я, прізвище, зарезервоване ім'я користувача, пароль, поштову адресу, дату створення користувача в системі та його країну проживання.

Зовнішнє зображення реєстраційної сторінки користувача зображено на рис. 3.10.



Detection Page
Home → Subscribe

Password:

Username: Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name:

Last name:

Email address:

Date joined: 2020-05-20 19:23:52

Age:

Country:

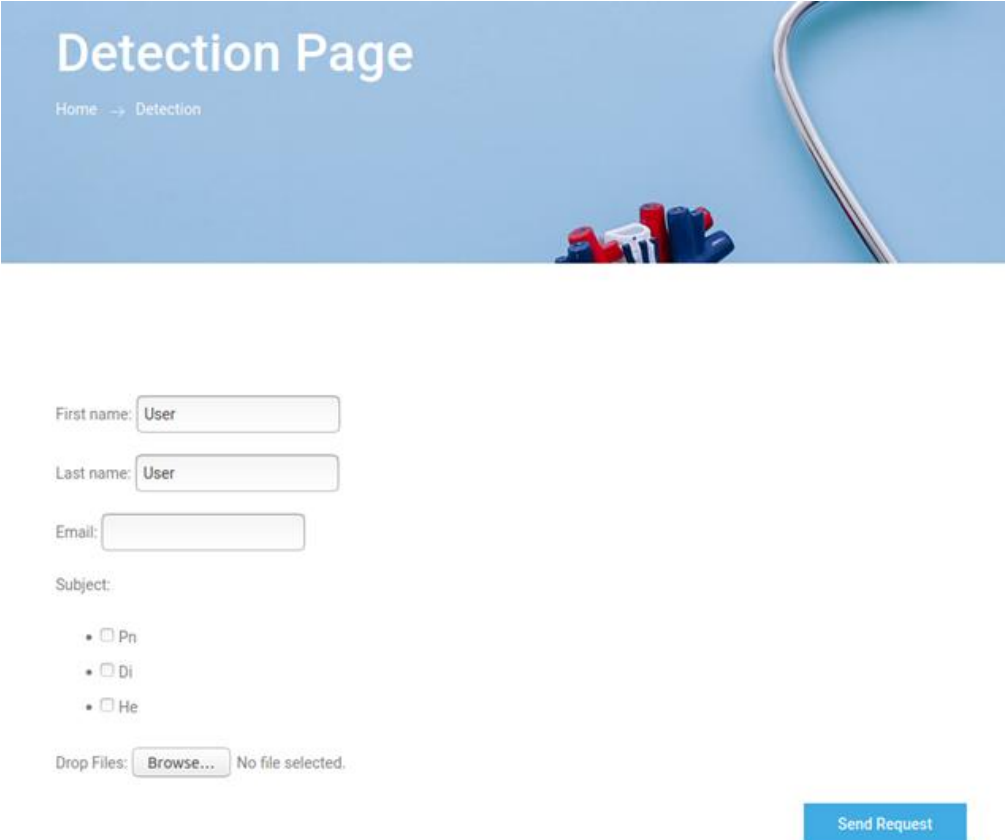
[Send Request](#)

Рисунок 3.10 – Вигляд сторінки реєстрації нового користувача у системі

Джерело: розроблено автором

Сторінка завантаження зображення для класифікації є водночас і сторінкою для авторизації користувача у системі. Для проведення даної операції використовуються поштова адреса, а також ім'я та прізвище користувача у системі. Далі знаходяться поля, що відповідають за завантаження потрібного зображення та вибір відповідного типу медичного діагностування, який користувач має намір здійснити.

Зовнішнє зображення сторінки завантаження зображення показано на рис. 3.11.



Detection Page
Home → Detection

First name:

Last name:

Email:

Subject:

- ☐ Pn
- ☐ Di
- ☐ He

Drop Files: No file selected.

Рисунок 3.11 – Вигляд сторінки авторизації та завантаження зображення

Джерело: розроблено автором

У разі, якщо користувач є зареєстрованим у системі, то далі, після обчислення результатів роботи внутрішнього алгоритму класифікації, користувач буде автоматично переведений на сторінку виводу та візуалізації результатів.

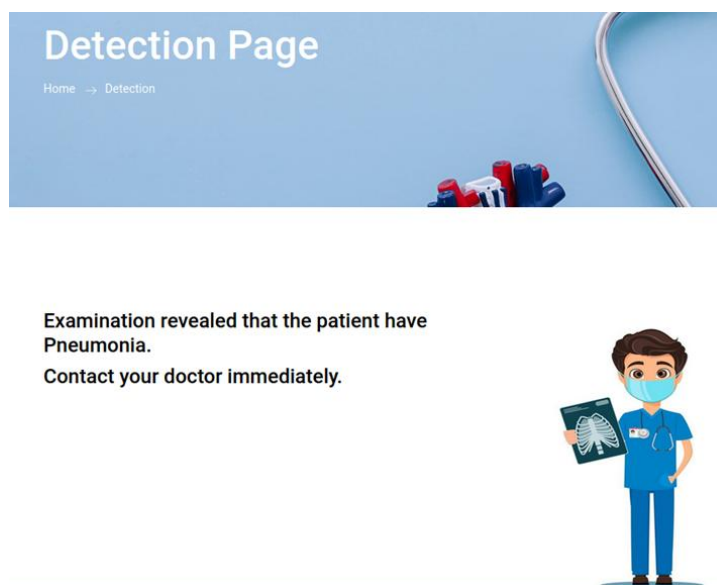


Рисунок 3.12 – Вигляд сторінки виведення результатів діагностики на прикладі задачі діагностування пневмонії

Джерело: розроблено автором

3.5 Аналіз отриманих результатів

Усі класифікаційні алгоритми, що були реалізовані під час виконання роботи, розроблялися із використанням пакету для глибокого навчання на мові програмування Python — Keras. Даний пакет підтримує можливість візуалізації процесу навчання моделі, що дає змогу відслідковувати усі ітерації, тим самим обираючи найбільш стійкий алгоритм розв’язання задачі. Розглянемо основні показники (див. рис. 3.13 – 3.14), що застосовувались на даному етапі оцінки моделей, на прикладі класифікатора діабетичної ретинопатії за зображенням сітчатки ока.

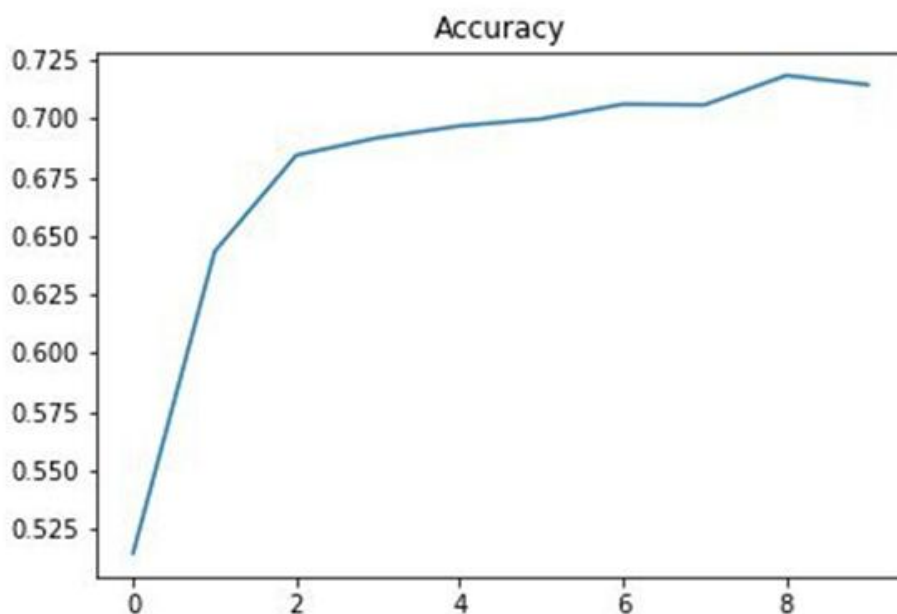


Рисунок 3.13 – Залежність валідаційної метрики – точність, від кількості ітерацій

Джерело: розроблено автором

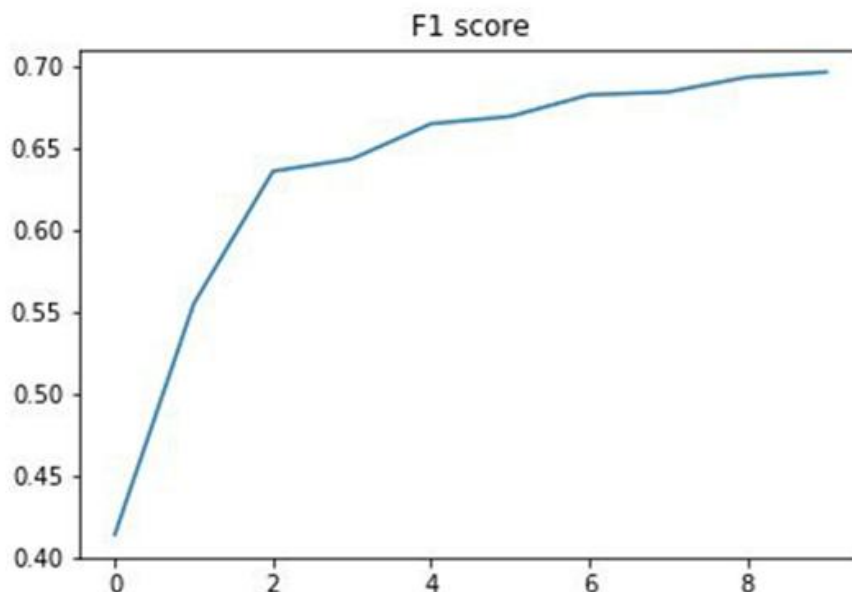


Рисунок 3.14 – Залежність валідаційної метрики – F1, від кількості ітерацій

Джерело: розроблено автором

Окрім, використовувались й інші засоби для візуалізації результатів навчання класифікаційного алгоритму. Зокрема, широко застосовувалась статистика Колмогорова-Смірнова, візуалізація якою була реалізована із застосуванням пакету `scikit-plot` (див.рис.3.15). Дана статистика будується шляхом побудови кумулятивних функцій розподілу для кожного із класів в залежності від прогнозної ймовірності моделі, якість якої тестується. Далі, при кожному значенні ймовірності обчислюється відстань між даними функціями розподілу і шукається точка, в якій дана відстань буде максимальною [21, ст. 231]. Таким чином, у шуканій точці побудований класифікатор буде мати найбільшу узагальнюючу здатність.

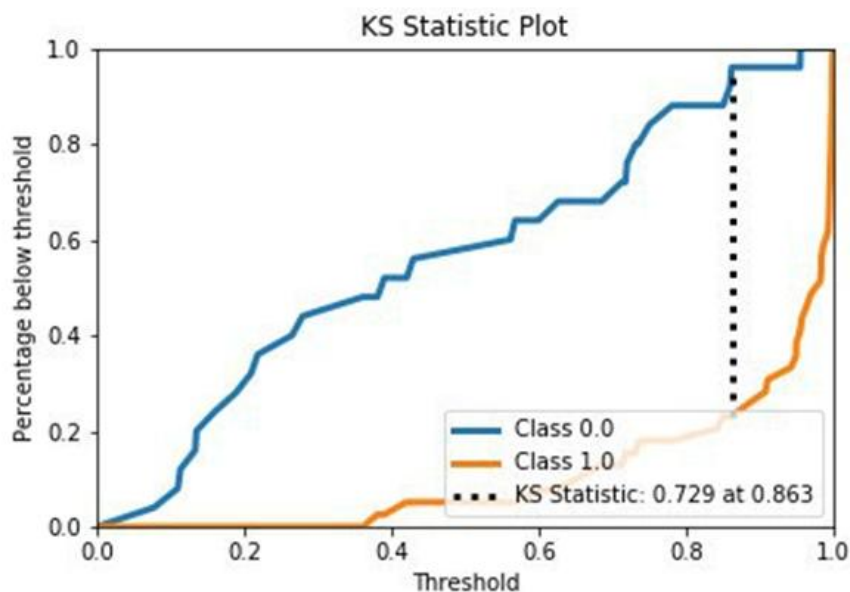


Рисунок 3. 15 – Графік залежності кумулятивних функцій розподілу в залежності від прогнозної ймовірності

Джерело: розроблено автором

Отже, застосовуючи дані методи для оцінки якості роботи моделі під час навчання, були отримані такі результати на валідаційній вибірці, що наведені у таблиці нижче.

Таблиця 3.1 — Результати оцінювання моделей

Тип моделі	AUC	F1	Accuracy
Модель класифікації пневмонії	0.81	0.84	0.80
Модель класифікації діабетичної ретинопатії	0.8	0.7	0.75
Модель класифікації крововиливу у мозок	0.7	0.68	0.71

Джерело: розроблено автором

Отже, за результатами оцінювання моделей на відповідних валідаційних вибірках, було показано, що побудовані моделі є досить стійкими та мають гарну узагальнюючу здатність.

3.6 Висновки до розділу 3

У даному розділі було аргументовано вибір мови програмування, а також основних фреймворків та програмних пакетів для вирішення поставленої задачі. Проведено детальний опис, а також базовий експлоративний аналіз вхідних даних, що використовувались при побудові відповідних класифікаторів.

Розглянуто основні архітектурні елементи програмного продукту, проведено його розбиття на рівні абстракції, проаналізовано алгоритм отримання результатів роботи сервісу. Також був описаний та візуалізований користувацький інтерфейс, а також наведені інструкції з експлуатації сервісу.

Проведено опис та аналіз отриманих результатів роботи класифікаторів, означено подальші перспективи їх розвитку та сервісу як програмного продукту в цілому.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО – ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

4.1 Обґрунтування функцій та параметрів програмного продукту

Постановка задачі проектування зводиться до розробки програмного продукту, який проводитиме медичне діагностування захворювань пацієнтів-користувачів платформи. Він призначений для використання в середовищах що підтримують технології машинного навчання на Python та фреймворку Django.

З огляду на таке визначення основної мети розробки продукту можна зазначити, що головними компонентами (функціями-складниками) будуть слугувати:

а) F_1 – вибір мови програмування:

- 1) мова програмування Python;
- 2) мова програмування C++;

б) F_2 – вибір типу обчислювальних ресурсів:

- 1) CPU;
- 2) GPU;

в) F_3 – вибір платформи для побудови сервісу:

- 1) фреймворк Django;
- 2) фреймворк Crow;

г) F_4 – вибір підходу до проведення медичної діагностики на основі зображень:

- 1) методи машинного навчання;
- 2) побудова емпіричних правил на вхідне зображення.

Усі можливі варіанти застосування зазначених вище функцій позначено на морфологічній карті системи, що відображає рисунок 4.1.

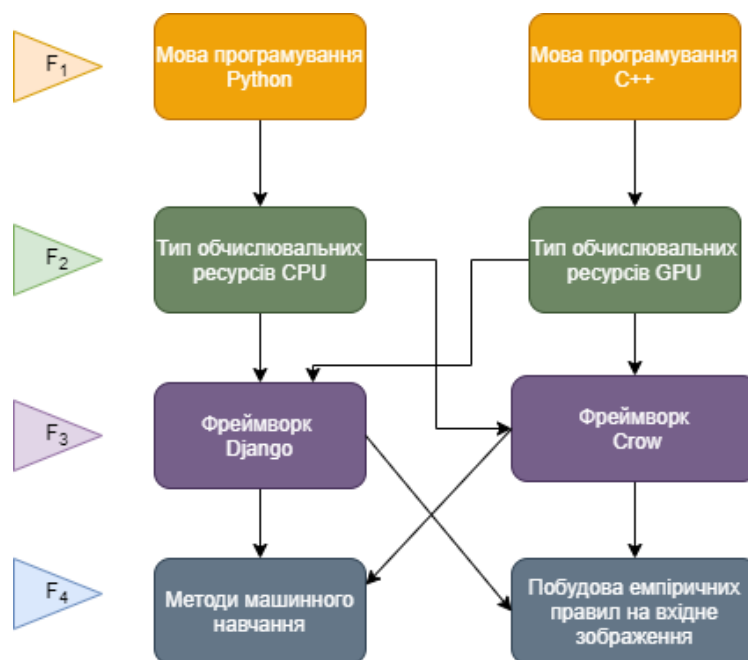


Рисунок 4.1 – Морфологічна карта системи

Джерело: розроблено автором

На основі побудованої карти було сформовано позитивно-негативну матрицю варіантів основних функцій. Детальний опис їх переваг та недоліків виведено в таблиці 4.1.

Таблиця 4.1 – Позитивно-негативна матриця

Основні функції	Варіанти	Переваги	Недоліки
F_1	А	Лаконічність запису коду та легкість його реалізації, наявність інструментарію для машин навчання	Повільне виконання коду
	Б	Можливість оптимізації базових процесів, висока швидкодія виконання коду	Схильність до великої кількості залежностей в коді, великий обсяг його написання, мало спеціалізованих бібліотек і важкість їх використання
F_2	А	Велика функціональність	Обмежена кількість потоків
	Б	Довільна кількість потоків	Витрачає велику кількість енергії
F_3	А	Наявність ORM архітектури, підтримка стандартної бази даних SQLite за базових налаштувань, наявність побудованої реалізації багатьох базових процесів	Повільний у роботі, велика кількість залежності у фреймворку
	Б	Простіший і швидший у використанні,	Відсутність ORM архітектури, потреба у реалізації багатьох базових процесів власноруч

Продовження таблиці 4.1

Основні функції	Варіанти	Переваги	Недоліки
F_4	А	Робастне, інваріантне до вхідного зображення, висока точність одержаних результатів	Складність реалізації, потреба у великій кількості обчислювальних ресурсів
	Б	Простота у реалізації, відсутність потреби у використанні великої кількості програмних засобів	Відсутність інваріантності: при суттєвій зміні вхідного зображення необхідно будувати нове емпіричне правило

Джерело: розроблено автором

Для проведення детальної характеристики програмного продукту нами обрані параметри та їх значення (табл. 4.2.)

Таблиця 4.2 – Основні параметри програмного продукту та градація їх значень

Назва Параметра	Умовні позначення	Одиниці вимірювання	Абсолютне значення параметру		
			краще	середнє	Гірше
Швидкість роботи програмного коду	X_1	Мс	100	500	1500
Об'єм пам'яті необхідний для відпрацювання алгоритму	X_2	Гб	8	16	128
Час обробки запиту на розпізнавання вихідного зображення	X_3	С	45	60	90
Точність прогнозування	X_4	доля одиниці	0,9	0,8	0,6

Джерело: розроблено автором

Значення будь-якого параметра оцінюється в балах: краще значення — в 10 балів, гірше — в 1 бал. За даними таблиці 4.2 будуються графічні характеристики параметрів (рис. 4.2).

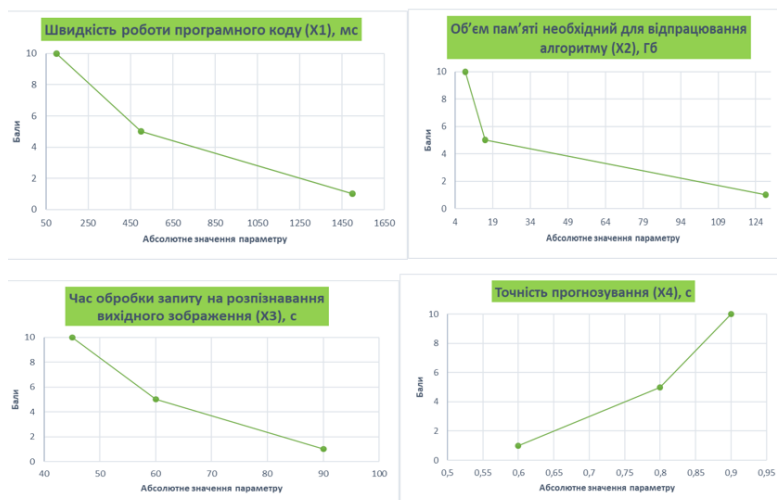


Рисунок 4.2 – Значення параметрів, що характеризують програмний продукт

Джерело: розроблено автором

Наступним кроком після аналізу значень параметрів буде формування групи експертів, що має на меті оцінити ступінь важливості кожного параметру за методом попарного порівняння. Зазначимо, що ранги варіюватимуться від 1 до 4.

Результати експертного ранжування подані в таблиці 4.3.

Таблиця 4.3 – Результати ранжування головних параметрів програмного продукту

Параметр	Ранг параметра за оцінкою експерта								Сума рангів R_i	Δ_i	Δ_i^2
	Експ1	Експ2	Експ3	Експ4	Експ5	Експ6	Експ7	Експ8			
X_1	3	1	2	1	2	2	1	2	14	-6	36
X_2	1	2	1	2	1	1	2	1	11	-9	81
X_3	2	4	3	4	4	3	3	3	26	6	36
X_4	4	3	4	3	3	4	4	4	29	9	81
Загальне значення	10	10	10	10	10	10	10	10	80	0	234

Джерело: розроблено автором

За найменший ранг приймається 1, найбільший – 4 бали.

Проведення попарного порівняння всіх параметрів та розрахунок коефіцієнтів їх вагомості подаються у таблиці 4.4.

Таблиця 4.4 – Попарне порівняння параметрів

Параметри	Експ1	Експ2	Експ3	Експ4	Експ5	Експ6	Експ7	Експ8	Кінцева оцінка	Числове значення
X_1 і X_2	>	<	>	<	>	>	<	>	>	1,5
X_1 і X_3	>	<	<	<	<	<	<	<	<	0,5
X_1 і X_4	<	<	<	<	<	<	<	<	<	0,5
X_2 і X_3	<	<	<	<	<	<	<	<	<	0,5
X_2 і X_4	<	<	<	<	<	<	<	<	<	0,5
X_3 і X_4	<	>	<	>	>	<	<	<	<	0,5

Джерело: розроблено автором

За одержаними результатами експертного ранжування розрахуємо коефіцієнт узгодженості (конкордації):

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 234}{8^2(4^3 - 4)} = 0,73, \quad (4.1)$$

де N – кількість експертів;

n – кількість параметрів.

За розрахованими результатами співвідношення (4.1) може зазначити, що так як коефіцієнт конкордації більше нормативного ($W = 0,73 > W_k = 0,67$), то результати є достовірними.

Розрахунок вагомості параметрів наведено в табл. 4.5

Таблиця 4.5 – Коефіцієнти вагомості параметрів

Параметри і	Параметри j				1 ітерація		2 ітерація		3 ітерація	
	X ₁	X ₂	X ₃	X ₄	b ¹ _i	K ¹ _{Bi}	b ² _i	K ² _{Bi}	b ³ _i	K ³ _{Bi}
X ₁	1	1,5	0,5	0,5	3,5	0,219	12,3	0,208	62,5	0,268
X ₂	0,5	1	0,5	0,5	2,5	0,156	9,25	0,157	34,13	0,146
X ₃	1,5	1,5	1	0,5	4,5	0,281	16,3	0,275	59,13	0,253
X ₄	1,5	1,5	1,5	1	5,5	0,344	21,3	0,360	77,88	0,333
Всього					16	1	59	1	233,6	1

Джерело: розроблено автором

Враховуючи дані з порівнянь варіантів реалізацій функцій можна виключити з реалізацій функцій наступні варіанти: $F_1(b)$, $F_3(b)$ та $F_4(b)$. Залишаються наступні варіанти:

а) $F_1(a) \rightarrow F_2(a) \rightarrow F_3(a) \rightarrow F_4(a)$;

б) $F_1(a) \rightarrow F_2(b) \rightarrow F_3(a) \rightarrow F_4(a)$.

Результати за кожним із варіантом виведені в таблиці 4.6.

Таблиця 4.6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
$F_1(x_1)$	A	250	8	0,268	2,144
$F_2(x_2)$	A	42	4	0,146	0,584
	B	17	6	0,146	0,876
$F_3(x_3)$	A	51	8	0,253	2,024

$F_4(x_4)$	A	0,87	8	0,333	2,664
------------	---	------	---	-------	-------

Джерело: розроблено автором

Визначимо рівень якості для кожного з варіантів окремо:

$$K_{K1} = 2,144 + 0,584 + 2,024 + 2,664 = 7,416, \quad (4.2)$$

$$K_{K2} = 2,144 + 0,876 + 2,024 + 2,664 = 7,708 \quad (4.3)$$

Як видно з розрахунків (4.2) та (4.3), кращим є другий варіант, для якого коефіцієнт технічного рівня має найбільше значення.

4.2 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Усі варіанти включають в себе три однакових завдання:

- 1) розробка класифікаційного алгоритму методами машинного навчання;
- 2) розробка програмної оболонки на мові Python;
- 3) розробка сервісу за допомогою фреймворку Django.

Також кожний з варіантів має додаткове завдання, що є реалізацією розгалужених варіантів розробки незалежного модуля. Далі наведено варіанти додаткового завдання:

- 4.1) тип обчислювальних ресурсів CPU;
- 4.2) тип обчислювальних ресурсів GPU.

У варіанті 1 присутнє додаткове завдання під номерами 4.1, а у варіанті 2 – завдання під номерами 4.2.

За ступенем новизни до групи Б відноситься завдання 2, 3 та 4.1, а до групи В відносяться завдання 1 та 4.2. За складністю алгоритмів до групи 1 відносяться завдання 1, 2, 3 та 4.1, а до групи 3 відноситься завдання 4.2. Для реалізації завдання 1 та 4.2 використовується інформація у вигляді даних (БД), а для завдання 2, 3 та 4.1

– нормативно-довідкова інформація (НДІ). Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Спираючись на норми розрахункового часу визначимо трудомісткість. Вона складає для першого завдання $T_p=43$ людино-днів. Поправочний коефіцієнт інформація у вигляді даних (K_{Π}) становить 0,68, коефіцієнт, який враховує складність контролю вхідної та вихідної інформації (K_{CK}) – 1,07, коефіцієнт використання стандартних модулів та прикладних програм (K_{CT}) – 0,7, а коефіцієнт стандартного математичного забезпечення ($K_{CT.M}$) – 1,3.

Визначивши усі необхідні поправочні коефіцієнти та норму розрахункового часу, можемо розрахувати загальну трудомісткість програмування першого завдання (людино-днів):

$$T_1 = 43 * 0,68 * 1,07 * 0,7 * 1,3 = 28,47$$

Проведемо аналогічні розрахунки:

а) для завдання 2 (складність – 1, новизна – Б, інформація – НДІ) $T_p=64$; $K_{\Pi}=1,021$; $K_{CK}=1,16$, $K_{CT}=0,8$, $K_{CT.M}=1,6$:

$$T_2 = 64 * 1,021 * 1,16 * 0,8 * 1,6 = 97,02$$

б) для завдання 3 (складність – 1, новизна – Б, інформація – НДІ) $T_p=64$; $K_{\Pi}=1,021$; $K_{CK}=1,08$, $K_{CT}=0,7$, $K_{CT.M}=1,4$:

$$T_3 = 64 * 1,021 * 1,08 * 0,7 * 1,4 = 69,16$$

в) для завдання 4.1 (складність – 1, новизна – Б, інформація – БД) $T_p=64$; $K_{\Pi}=1,01$; $K_{CK}=1,07$, $K_{CT}=0,8$, $K_{CT.M}=1,4$:

$$T_{4.1} = 64 * 1,01 * 1,07 * 0,8 * 1,4 = 77,46$$

г) для завдання 4.2 (складність – 3, новизна – В, інформація – БД) $T_p = 12$; $K_n = 0,50$; $K_{ск} = 1$, $K_{ст} = 0,8$, $K_{ст.м} = 1,2$:

$$T_{4.2} = 12 * 0,5 * 1 * 0,8 * 1,2 = 5,76$$

Визначимо повну трудомісткість варіантів(людино-днів):

$$T_1 = (28,47 + 97,02 + 69,16 + 77,46) * 8 = 2176,88, \quad (4.4)$$

$$T_2 = (28,47 + 97,02 + 69,16 + 5,76) * 8 = 1605,44 \quad (4.5)$$

Отже, найбільш високу трудомісткість має перший варіант.

В розробці беруть участь один бекенд-розробник з окладом 21440 грн. та один інженер машинного навчання з окладом 40000 грн. Визначимо середню заробітну плату за годину (грн):

$$C_q = \frac{21440 + 40000}{2 * 21 * 8} = 182,86 \quad (4.6)$$

За результатами співвідношення (4.6) розрахуємо заробітну плату для кожного з варіантів реалізації(грн):

$$C_{зп_1} = 182,86 * 2176,88 * 1,2 = 477677,13, \quad (4.7)$$

$$C_{зп_2} = 182,86 * 1605,44 * 1,2 = 352284,91 \quad (4.8)$$

За результатами співвідношень (4.7) та (4.8) розрахуємо відрахування на соціальний внесок (22%) для кожного з варіантів реалізації (грн):

$$C_{від_1} = 477677,13 * 0,22 = 105088,97, \quad (4.9)$$

$$C_{від_2} = 352284,91 * 0,22 = 77502,68 \quad (4.10)$$

Тепер визначимо витрати на оплату однієї машино-години. Так як одна ЕОМ обслуговує одного програміста з окладом 21440 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо (грн):

$$C_{\Gamma} = 12 * M * K_3 = 12 * 21440 * 0,2 = 51456$$

Враховуючи додаткову заробітну плату витрати на оплату однієї машино-години становитимуть (грн):

$$C_{3\text{п}} = C_{\Gamma} * (1 + K_3) = 51456 * (1 + 0,2) = 61747,2 \quad (4.11)$$

Тепер за результатами співвідношення можемо визначити відрахування на соціальний внесок (грн):

$$C_{\text{від}} = C_{3\text{п}} * 0,22 = 61747,2 * 0,22 = 13584,384 \quad (4.12)$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 24000 грн.:

$$C_A = K_{\text{ТМ}} * K_A * C_{\text{ПР}} = 1,15 * 0,25 * 24000 = 6900 \quad (4.13)$$

Витрати на ремонт та профілактику розраховуємо як (грн):

$$C_p = K_{\text{ТМ}} * C_{\text{ПР}} * K_p = 1,15 * 24000 * 0,05 = 1380 \quad (4.14)$$

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою (год):

$$T_{\text{ЕФ}} = (365 - 104 - 8 - 16) * 8 * 0,9 = 1706,4 \quad (4.15)$$

Витрати на оплату електроенергії розраховуємо за формулою (грн):

$$C_{\text{ЕЛ}} = 1706,4 * 0,156 * 0,2 * 1,75 = 93,17 \quad (4.16)$$

Накладні витрати розраховуємо за формулою (грн):

$$C_{\text{Н}} = C_{\text{ПР}} * 0,67 = 24000 * 0,67 = 16080 \quad (4.17)$$

За значеннями одержаних з формул (4.11) – (4.17) можемо розрахувати річні експлуатаційні витрати (грн):

$$\begin{aligned} C_{\text{екс}} &= 61747,2 + 13584,384 + 6900 + 1380 + 93,17 + 16080 = \\ &= 99784,754 \end{aligned} \quad (4.18)$$

За одержаними значеннями формул (4.18) та (4.15) можемо розрахувати собівартість однієї машино-години ЕОМ (грн/час):

$$C_{\text{м-г}} = \frac{C_{\text{екс}}}{T_{\text{еф}}} = \frac{99784,754}{1706,4} = 58,48 \quad (4.19)$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, використовуючи результат формули (4.19), обчислюється таким чином (грн):

а) для 1 варіанту:

$$C_{\text{м}_1} = 58,48 * 2176,88 = 127303,94 \quad (4.20)$$

б) для 2 варіанту:

$$C_{\text{м}_2} = 58,48 * 1605,44 = 93886,13 \quad (4.21)$$

Накладні витрати складають 67% від заробітної плати і для кожного з варіанту розраховується таким чином (грн):

а) для 1 варіанту:

$$C_{H_1} = 127303,94 * 0,67 = 85293,64 \quad (4.22)$$

б) для 2 варіанту:

$$C_{H_2} = 93886,13 * 0,67 = 62903,71 \quad (4.23)$$

Знаючи розрахунки за формулами (4.7) – (4.10) та (4.20) – (4.23) може знайти вартість розробки ПП для кожного з розглянутих варіантів (грн):

а) для 1 варіанту:

$$\begin{aligned} C_{ПП_1} &= 477677,13 + 105088,97 + 127303,94 + 85293,64 = \\ &= 795363,69 \end{aligned} \quad (4.24)$$

б) для 2 варіанту:

$$\begin{aligned} C_{ПП_2} &= 352284,91 + 77502,68 + 93886,13 + 62903,71 = \\ &= 586577,43 \end{aligned} \quad (4.25)$$

За значеннями, одержаними під час розрахунку формул (4.2, 4.3) та (4.24, 4.25), можемо обчислити коефіцієнт техніко-економічного рівня для кожного з варіантів:

а) для 1 варіанту:

$$K_{TEP_1} = \frac{7,416}{795363,69} = 0,93 * 10^{-5}$$

б) для 2 варіанту:

$$K_{\text{TEP}_2} = \frac{7,708}{586577,43} = 1,31 * 10^{-5}$$

Як бачимо, найбільш ефективним є другий варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{\text{TEP}_2} = 1,31 * 10^{-5}$.

4.3 Висновки до розділу 4

Отже, враховуючи всі дослідження, що описані вище, можна сказати, що 2 варіант реалізації є найбільш оптимальним зі сторони якісно-економічної оцінки. Його коефіцієнт техніко-економічного рівня складає $K_{\text{TEP}_2} = 1,31 * 10^{-5}$.

Розробка цього варіанту передбачає такі обов'язкові завдання як:

- а) розробка класифікаційного алгоритму методами машинного навчання;
- б) розробка програмної оболонки на мові Python;
- в) розробка сервісу за допомогою фреймворку Django.

Завдання для якого здійснювався вибір було обрано 2 варіант – тип обчислювальних ресурсів для використання – GPU.

ВИСНОВКИ

У даній бакалаврській дипломній роботі було сформульовано та розв'язано задачу проведення медичної діагностики пацієнта методами машинного навчання. У процесі виконання роботи було одержано результати, за якими можна такі висновки:

- а) було проаналізовано основні існуючі підходи до вирішення цієї проблеми. Проведено їх порівняльну характеристику. Здійснена формалізація задачі у термінах машинного навчання;
- б) здійснено аналіз основних методів машинного навчання для вирішення задачі класифікації зображень, розглянуто їх математичні основи, основні характеристики, а також переваги та недоліки. Проведено порівняльний аналіз даних методів, а також основних метрик оцінки якості вирішення поставленої задачі; За результатами даного аналізу у якості алгоритму вирішення задачі було обрано метод побудови конволютивних нейронних мереж, із відповідними модифікаціями під кожне конкретне захворювання;
- в) було створено програмний продукт, із використанням високорівневої мови програмування Python. Для побудови та валідації нейромереж використовувались відповідні модулі для роботи з методами машинного навчання: Keras, scikit-learn. Програмний продукт є сервісом, що побудований у вигляді API із використанням фреймворку Django. Реалізовано можливість авторизації користувача та збереження результатів роботи програми до бази даних. У якості сховища даних використовувалось стандартне рішення фреймворку Django – SQLite.
- г) значущість створеного продукту полягає у покращенні існуючого процесу проведення медичного діагностування пацієнтів на такі захворювання як пневмонія, крововилив у мозок та діабетична ретинопатія. Побудований сервіс може бути використаний як лікарями, так і пацієнтами для пришвидшення процесу проведення відповідного діагностування. Результати роботи можуть бути покращені шляхом використання більшого об'єму даних

для тренування, збільшення кількості прихованих шарів нейронних мереж, але це, у свою чергу, потребує використання значно більшої кількості обчислювальних ресурсів.

ПЕРЕЛІК ПОСИЛАНЬ

1. DiMasi J. A., Hansen R. W., Grabowski H. G. The Price of Innovation: New Estimates of Drug Development Costs. *J Health Econ.* 2003. Vol. 22, No. 2. P. 151–185.
2. Ekins S., Puhl C. A., Kimberley M. Z., Thomas R. L., Daniel P. R. Clark Exploiting machine learning for end-to-end drug discovery and development. *Nature Materials.* 2019. Vol. 3. No. 18. P. 435-441.
3. Ekins S. Ascent of machine learning in medicine. *Nature Materials.* 2019. Vol. 3. No. 18. P. 407. URL: <https://www.nature.com/articles/s41563-019-0360-1> (Last accessed: 17.05.2020).
4. Офіційна сторінка BuoyHealth. URL: <https://www.buoyhealth.com/> (Last accessed: 17.05.2020).
5. Murugesan M. How do Machines Learn ?. *Abeyon.* 2018. URL: <https://abeyon.com/how-do-machines-learn/> (Last accessed: 17.05.2020).
6. Arbib. M. A. The handbook of brain theory and neural networks. Cambridge; London: A Bradford Book, 2003, 200 p.
7. Ribose Y. Machine Learning. *Machine Learning Journal.* 2018. URL: <https://zhuanlan.zhihu.com/p/36570080> (Last accessed: 17.05.2020).
8. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge, MA: The MIT Press Cambridge, 2018. 235 p.
9. Martin N. Convolutional Neural Networks. *SuperDataScience.* 2018. URL: <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1-convolution-operation> (Last accessed: 17.05.2020).
10. Lee F.F., William S., CS231n: Convolutional Neural Networks for Visual Recognition. Stanford, CA: Stanford University Press, 2020. URL: <https://cs231n.github.io/convolutional-networks/> (Last accessed: 17.05.2020).
11. Nelson D. What are Convolutional Neural Networks ?. *Unite.AI.* 2019. Vol. 10. No. 101. P. 100. URL: <https://www.unite.ai/what-are-convolutional-neural-networks/> (Last accessed: 17.05.2020).

12. Patterson J., Gibson A. Deep Learning: A Practitioner's Approach. New York: O'Reilly, 2014. 400 p.
13. Kanish S. A Quick Overview to the Transfer Learning and it's Significance in Real World Applications. *Medium*. 2019. Vol. 276. No 3. URL: <https://medium.com/towards-tech-intelligence/a-quick-overview-to-the-transfer-learning-and-its-significance-in-real-world-applications-790fb57debad> (Last accessed: 17.05.2020).
14. Müller A.C., Guido S. Introduction to Machine Learning with Python. New York: O'Reilly, 2016. 392 p.
15. Swapna V., Gaikwad A., Kulkarni V. Fault Diagnosis of Bearing of Electric Motor Using Wavelet Transform and Fault Classification Based on Support Vector Machine. *Advance Research in Electrical and Electronic Engineering*. 2015. Vol. 2. No. 5. P. 41-46.
16. Narkhede S. Understanding Confusion Matrix. *Towards Data Science*. 2018. Vol. 180. No 1. URL: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62> (Last accessed: 17.05.2020).
17. Chollet F. Deep Learning with Python. Shelter Island: Manning, 2017. 237 p.
18. Nowak S., Grzybowski A. Review of the epidemiology of diabetic retinopathy. *Modern Retina*. 2018. Vol. 5. No. 3. P. 50.
19. Ford S. Death rates from community acquired pneumonia falling in NHS hospitals. *Nursing Times*. 2019. Vol. 6. No. 12. P. 10.
20. Syngo DE Brain Hemorrhage. *Siemens Healthineers*. URL: <https://www.siemens-healthineers.com/fr-ch/computed-tomography/options-upgrades/clinical-applications/syngo-de-brain-hemorrhage> (Last accessed: 17.05.2020).
21. Bruce P., Bruce A. Practical Statistics for Data Scientists. New York: O'Reilly, 2107. 409 p.

ДОДАТОК А ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

**Сервіс для проведення медичного діагностування
на основі машинного**
Service for conducting medical diagnosis based on
machine learning

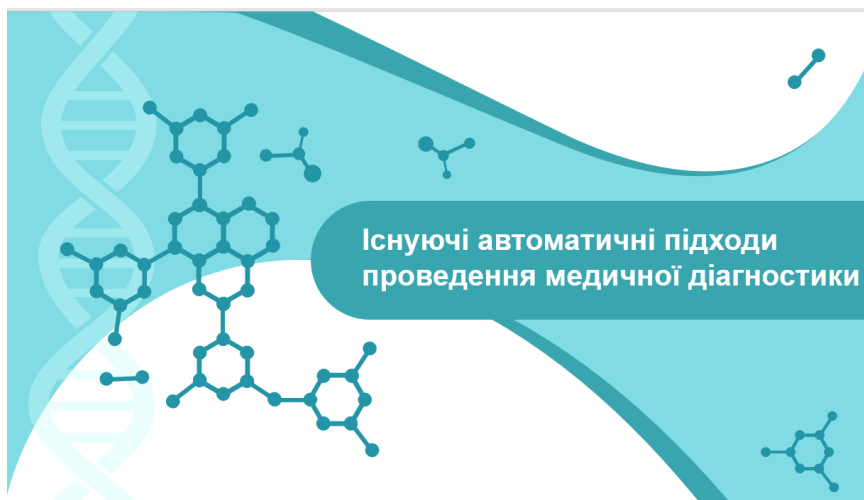
Науковий керівник
к.н.т., доцент
Дідковська М.В.

Виконав
студент групи КА-61
Климчук Ярослав

**Об'єкт, предмет, мета та
актуальність роботи**



Актуальність дипломної роботи



01 **Buoy Health**
Застосунок для діагностики симптомів хворого та проведення його лікування, що діє за принципом роботи чат-боту.

02 **Zebra Medical Vision**
Сервіс, розроблений із застосуванням алгоритмів глибокого навчання, зокрема глибоких нейронних мереж для асистування лікарів. Дана аналітична система на вхід отримує відскановані зображення захворювань головного мозку, легень, печінки, серцево-судинної системи, істотні проводить автоматичний аналіз даних зображень задля пошуку клінічного захворювання.

03 **PathAI**
Сервіс для надання допомоги лікарям при діагностуванні ракових пухлин

04 **BelevolentAI**
Сервіс для проведення персоналізації процесу прийняття рішень під час проведення медичного діагностування

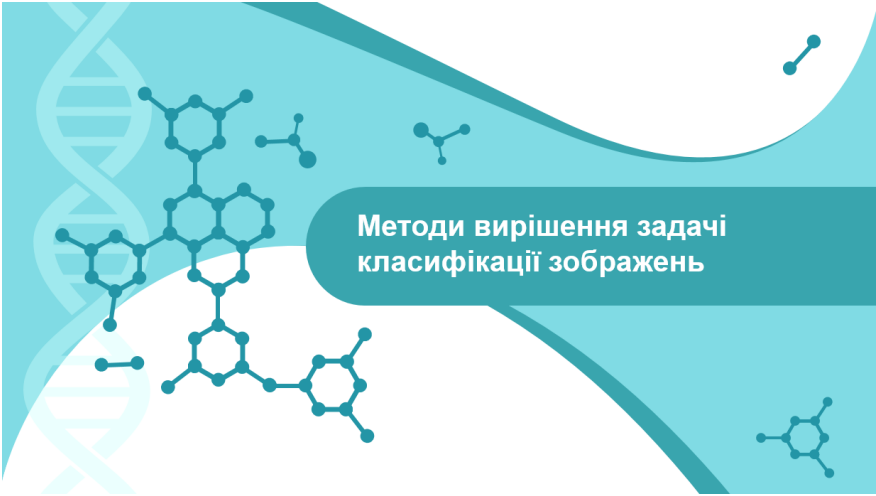
Характеристика існуючих підходів

Таблиця порівнянь

Висновки за платформами

Як бачимо, на даний момент **Zebra Medical Vision** та **BenevolentAI** вже перейшли стадію від стартапа до продуктової компанії, продукти якої використовуються в існуючих медичних системах провідних країн світу, що говорить про величезний потенціал застосування методів машинного навчання у медицині.

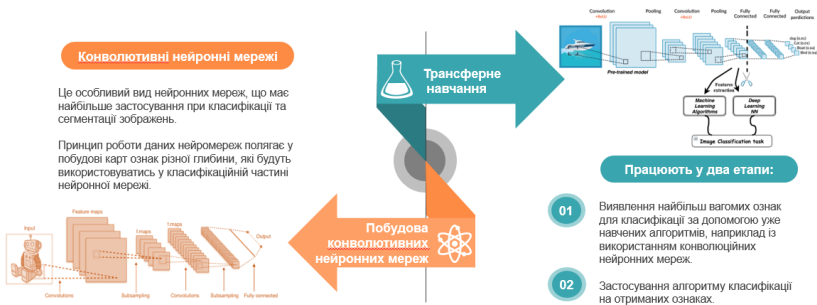
Критерій порівняння	Buoy Health	Zebra Medical Vision	Path AI	Benevolent AI
Наявність зовнішніх залежностей	+	-	-	-
Наявність міжнародних сертифікатів якості	-	+	-	+
Інтегрованість в існуючі медичні системи	-	+	-	+



Методи вирішення задачі класифікації зображень

Класифікація зображень

Тип задач, що розв'язуються методами машинного навчання з учителем.
Дана задача може бути вирішена такими методами:



Основні структурні елементи конволютивної нейронної мережі



Процес «згортки» відбувається у такі етапи:

- 1) вирахування скалярного добутку між ядром згортки та декодованою n -вимірною матрицею, що описує вхідне зображення;
- 2) «зсув» ядра згортки на відповідний крок так, щоб вирахувати її скалярний добуток з тією частиною матриці вхідного зображення, яка не була використана на попередньому кроці.

Шар субдискретизації

Цей шар використовується для зменшення розмірності карти ознак, яку ми отримали після шару згортки. Це робиться для того, щоб перешкодити перенаванчю нейромережі, а також вивченню локальних закономірностей, адже для подальшого вивчення зображення нам вже не потрібно знати точне місцезнаходження тієї чи іншої ознаки, важливим залишається лише факт її наявності на даній ділянці зображення.

Основні структурні елементи конволютивної нейронної мережі

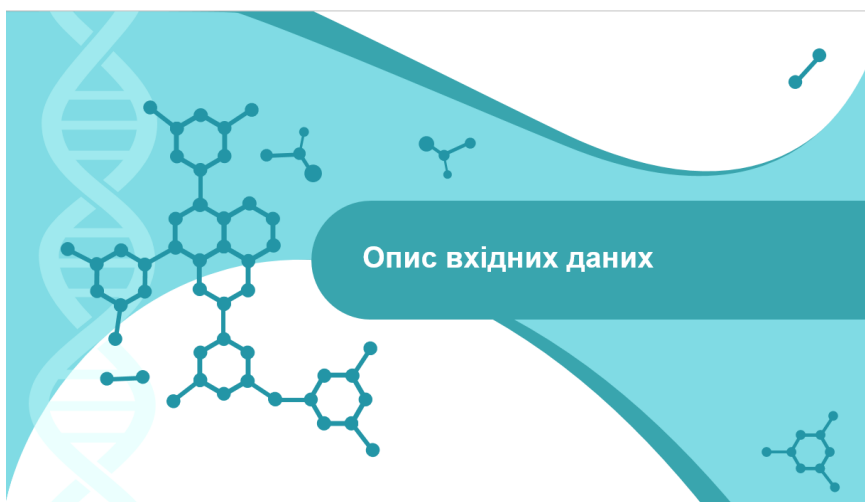


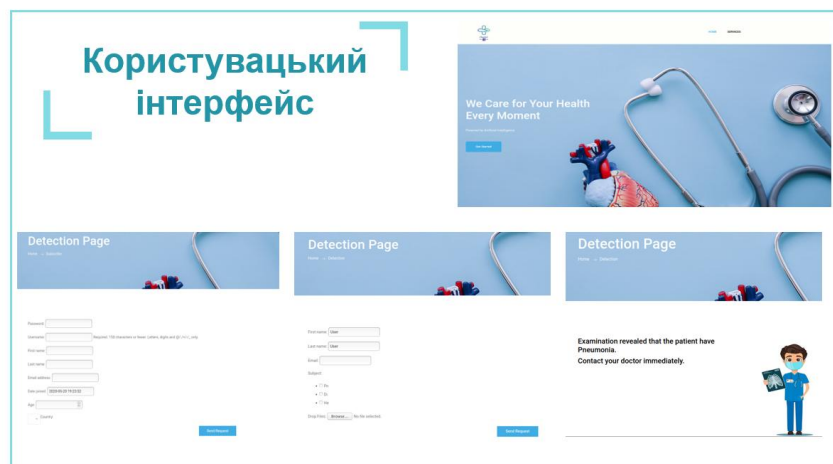
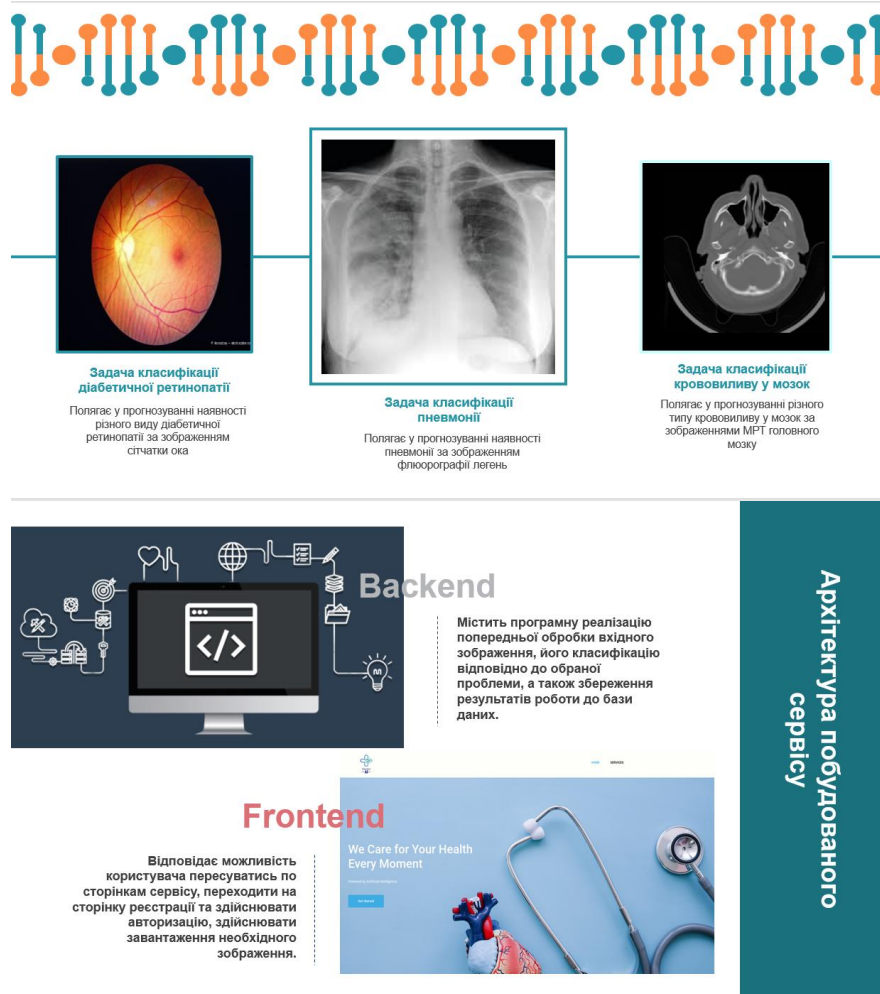
Шар Dropout

Цей метод використовується для того, щоб запобігти перенаванчю нейронної мережі та зменшити акцентування «уваги» нейронів на незнайомих, нетипових ознаках.

Повноз'язний шар

Працює як «звичайна» нейронна мережа: на вхід кожного нейрона подається лінійна комбінація ваг та вхідних даних (ознаки, що отримані зі згорткових шарів), виходом нейрона буде та ж лінійна комбінація, «пропущена» через функцію активації даного нейрона.





Результати роботи програмного продукту

Таблиця порівнянь

Висновки за результатами

Усі результати за моделями є стійкими та характеризують велику узагальнюючу здатність для кожної з побудованих моделей.

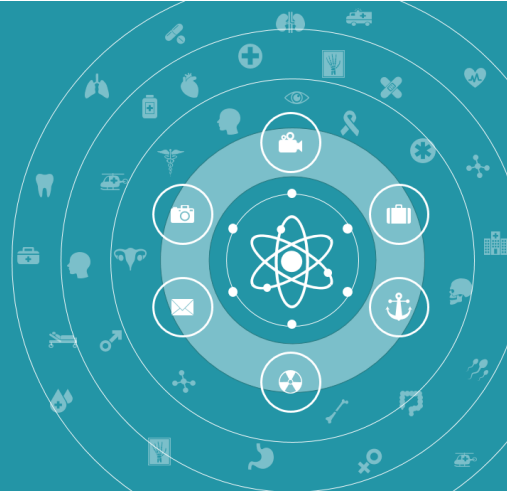
Оскільки модель класифікації пневмонії є моделлю бінарної класифікації, то її результати є найкращими серед усіх моделей.

Тип моделі	AUC	F1	Accuracy
Модель класифікації пневмонії	0,81	0,84	0,80
Модель класифікації діабетичної ретинопатії	0,80	0,70	0,75
Модель класифікації крововилив у мозок	0,70	0,68	0,71

Шляхи подальшого розвитку програмного продукту



Дякую
за
увагу !



ДОДАТОК Б КОД ПРОГРАМНОГО ПРОДУКТУ

```

from django.contrib import admin

from .models import General

admin.site.register(General)

from django.apps import AppConfig

class SubscribeConfig(AppConfig):
    name = 'medical_cases_recognition.apps.subscribe'

from django import forms
from .models import General, User

class UserForm(forms.ModelForm):

    class Meta:
        model = User
        exclude = ['']

    def __init__(self, *args, **kwargs):
        super(UserForm, self).__init__(*args, **kwargs)
        self.fields['last_login'].required = False
        self.fields['last_login'].widget = forms.HiddenInput()
        self.fields['user_permissions'].required = False
        self.fields['user_permissions'].widget = forms.HiddenInput()
        self.fields['groups'].required = False
        self.fields['groups'].widget = forms.HiddenInput()
        self.fields['is_staff'].required = False
        self.fields['is_staff'].widget = forms.HiddenInput()
        self.fields['is_superuser'].required = False
        self.fields['is_superuser'].widget = forms.HiddenInput()
        self.fields['is_active'].required = False
        self.fields['is_active'].widget = forms.HiddenInput()

class GeneralForm(forms.ModelForm):

    class Meta:

```

```
model = General
exclude = ['']
```

```
def __init__(self, *args, **kwargs):
    super(GeneralForm, self).__init__(*args, **kwargs)
    self.fields['user'].required = False
    self.fields['user'].widget = forms.HiddenInput()
```

```
from django.db import models
from django.contrib.auth.models import AbstractBaseUser
from django_countries.fields import CountryField
from django.core.mail import send_mail
from django.contrib.auth.models import PermissionsMixin
from django.contrib.auth.models import User
```

```
class General(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE, null=True)
    age = models.IntegerField(verbose_name='age', max_length=200)
    country = CountryField(verbose_name='country', multiple=True)
```

```
from django.conf.urls import url
from .views import ResponseForSubscribe
```

```
urlpatterns = [
    url(r'^$', ResponseForSubscribe, name='subscribe')
]
```

```
from django.shortcuts import render
from .forms import GeneralForm, UserForm
from django.http import HttpResponse
```

```
def ResponseForSubscribe(request):
    if request.method == 'POST':
        user_form = UserForm(request.POST, request.FILES)
        general_form = GeneralForm(request.POST, request.FILES)

        if general_form.is_valid() and user_form.is_valid():
            user_form.save()
            general_form.save()

            return render(request, "subscribe_result.html")
    else:
```

```

    return HttpResponse('subscribe failed with errors: {}'.format(user_form.errors))
else:
    raw_user_form = UserForm()
    raw_general_form = GeneralForm()
    return render(request, "subscribe.html",
                  {'user_form': raw_user_form,
                   'general_form': raw_general_form
                  })

```

```

from django.contrib import admin
from django.apps import AppConfig

```

```

class ServicesConfig(AppConfig):
    name = 'medical_cases_recognition.apps.services'

```

```

from django.db import models

```

```

from django.conf.urls import url
from .views import ResponseForServices

```

```

urlpatterns = [
    url(r'^$', ResponseForServices, name='services')
]
from django.shortcuts import render

```

```

def ResponseForServices(request):
    return render(request, "services.html", locals())
from django.contrib import admin

```

```

from django.apps import AppConfig

```

```

class HomeConfig(AppConfig):
    name = 'medical_cases_recognition.apps.home'

```

```

from django.db import models

```

```

from django.conf.urls import url
from .views import ResponseForHome

```

```

urlpatterns = [
    url(r'^$', ResponseForHome, name='home')
]

```

```
from django.shortcuts import render
```

```
def ResponseForHome(request):  
    return render(request, "index.html", locals())
```

```
from django.contrib import admin
```

```
from .models import Detection
```

```
admin.site.register(Detection)
```

```
from django.apps import AppConfig
```

```
class DetectionConfig(AppConfig):
    name = 'medical_cases_recognition.apps.detection'
```

```
from decouple import config
```

```
class ModelsConfig:
    MDL_CHEST_X_RAYS = config('MDL_CHEST_X_RAYS')
    WEIGHTS_MDL_CHEST_X_RAYS =
config('WEIGHTS_MDL_CHEST_X_RAYS')
    MDL_DIABETIC = config('MDL_DIABETIC')
    WEIGHTS_DIABETIC = config('WEIGHTS_DIABETIC')
    MDL_HEMORRHAGE = config('MDL_HEMORRHAGE')
    WEIGHTS_HEMORRHAGE = config('WEIGHTS_HEMORRHAGE')
    CUTOFF = config('CUTOFF')
```

```
from tensorflow.keras.models import model_from_json
```

```
from ...tools.logging_config import logger
```

```
diabet_decisions_mapping = {0: 'Mild',  
                             1: 'Severe',  
                             2: 'Proliferate DR',  
                             3: 'No DR',  
                             4: 'Moderate'  
                             }
```

[illegible]


```

2: 'Intraventricular Hemorrhage',
3: 'Subarachnoid Hemorrhage',
4: 'Subdural Hemorrhage',
5: 'No Hemorrhage'
}

```

```

def get_model mdl, weights, path_mdl, path_weights):
    json_file = open(path_mdl+mdl, 'r')
    loaded_model_json = json_file.read()
    json_file.close()
    loaded_model = model_from_json(loaded_model_json)
    loaded_model.load_weights(path_weights+weights)
    loaded_model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

```

```

    return loaded_model

```

```

def predict_x_ray(mdl, img, cutoff, decision):
    prediction = mdl.predict(img)[0][0]

    logger.info('Predicted probability: {}'.format(prediction))

    if prediction >= cutoff:
        return decision
    else:
        return 'No {}'.format(decision)

```

```

def predict_categorical_problem(mdl, img, decisions_mapping):
    prediction = mdl.predict(img).argmax()
    mapped_prediction = decisions_mapping.get(prediction)

    logger.info('Predicted probability: {}'.format(mdl.predict(img).max()))

    return mapped_prediction

```

```

from tensorflow.keras.models import model_from_json
from ...tools.logging_config import logger

```

```

diabet_decisions_mapping = {0: 'Mild',
1: 'Severe',
2: 'Proliferate DR',

```

```

3: 'No DR',
4: 'Moderate'
}

```

```

hemorrhage_decisions_mapping = {0: 'Epidural Hemorrhage',
                                1: 'Intraparenchymal Hemorrhage',
                                2: 'Intraventricular Hemorrhage',
                                3: 'Subarachnoid Hemorrhage',
                                4: 'Subdural Hemorrhage',
                                5: 'No Hemorrhage'
                                }

```

```

def get_model mdl, weights, path_md1, path_weights):
    json_file = open(path_md1+mdl, 'r')
    loaded_model_json = json_file.read()
    json_file.close()
    loaded_model = model_from_json(loaded_model_json)
    loaded_model.load_weights(path_weights+weights)
    loaded_model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

    return loaded_model

```

```

def predict_x_ray(mdl, img, cutoff, decision):
    prediction = mdl.predict(img)[0][0]

    logger.info('Predicted probability: {}'.format(prediction))

    if prediction >= cutoff:
        return decision
    else:
        return 'No {}'.format(decision)

```

```

def predict_categorical_problem(mdl, img, decisions_mapping):
    prediction = mdl.predict(img).argmax()
    mapped_prediction = decisions_mapping.get(prediction)

    logger.info('Predicted probability: {}'.format(mdl.predict(img).max()))

    return mapped_prediction

```

```

from django.db import models
from multiselectfield import MultiSelectField
from django.utils import timezone

```

```

class Detection(models.Model):
    OPTIONS = (
        ('Pneumonia', "Pn"),
        ('Diabetic', "Di"),
        ('Hemorrhage', "He")
    )
    first_name = models.CharField(verbose_name='first name', default='User',
max_length=500)
    last_name = models.CharField(verbose_name='last name', default='User',
max_length=500)
    email = models.EmailField(verbose_name='email')
    subject = MultiSelectField(verbose_name='subject', choices=OPTIONS)
    img_to_detect = models.ImageField(verbose_name='Drop Files', upload_to='images',
null=True)
    created = models.DateTimeField(verbose_name='created date',
default=timezone.now())

```

```

class Results(models.Model):
    email = models.EmailField(verbose_name='email', null=True)
    result = models.CharField(verbose_name='result', max_length=200, null=True)
    created = models.DateTimeField(verbose_name='result created date',
default=timezone.now(), null=True)

```

```

from .detection import (predict_x_ray, predict_categorical_problem, get_model,
diabet_decisions_mapping,
                        hemorrhage_decisions_mapping)
from .base import ModelsConfig
from keras.preprocessing.image import load_img, img_to_array
import numpy as np
from ...settings import MEDIA_ROOT

```

```

MODELS_MAPPING = {'Pneumonia': ModelsConfig.MDL_CHEST_X_RAYS,
                  'Diabetic': ModelsConfig.MDL_DIABETIC,
                  'Hemorrhage': ModelsConfig.MDL_HEMORRHAGE
                  }

```

```

WEIGHTS_MAPPING = {'Pneumonia':
ModelsConfig.WEIGHTS_MDL_CHEST_X_RAYS,

```

```

    'Diabetic': ModelsConfig.WEIGHTS_DIABETIC,
    'Hemorrhage': ModelsConfig.WEIGHTS_HEMORRHAGE
}

```

```

CUTOFFS_MAPPING = {'Pneumonia': ModelsConfig.CUTOFF
}

```

```

TARGET_SIZES_MAPPING = {'Pneumonia': (128, 128),
    'Diabetic': (128, 128),
    'Hemorrhage': (224, 224)
}

```

```

CATEGORIES_MAPPING = {'Diabetic': diabet_decisions_mapping,
    'Hemorrhage': hemorrhage_decisions_mapping
}

```

```

def preprocess(request, model, cutoff, problem, target_size, categorical_mapping=None):
    img_path = request.FILES['img_to_detect'].name
    img = img_to_array(load_img(MEDIA_ROOT + '/images/' + img_path,
target_size=target_size))
    img = np.expand_dims(img, axis=0)
    if problem == 'Pneumonia':
        result = predict_x_ray(model, img, float(cutoff), problem)
    else:
        result = predict_categorical_problem(model, img, categorical_mapping)

    return result

```

```

def define_problem(request):
    problem = request.POST.get('subject')
    mdl = MODELS_MAPPING.get(problem)
    weights = WEIGHTS_MAPPING.get(problem)
    cutoff = CUTOFFS_MAPPING.get(problem)
    target_size = TARGET_SIZES_MAPPING.get(problem)
    categorical_mapping = CATEGORIES_MAPPING.get(problem, None)

    return mdl, weights, cutoff, problem, target_size, categorical_mapping

```

```

def make_prediction(request):
    mdl, weights, cutoff, problem, target_size, categorical_mapping =
define_problem(request)
    model = get_model(mdl, weights, 'medical_cases_recognition/models/',

```

```
'medical_cases_recognition/weights/')
```

```
    result = preprocess(request, model, cutoff, problem, target_size, categorical_mapping)
```

```
    return result, problem
```

```
from django.conf.urls import url
```

```
from .views import ResponseForDetection
```

```
from ..subscribe.views import ResponseForSubscribe
```

```
urlpatterns = [
```

```
    url(r'^', ResponseForDetection, name='detection'),
```

```
    url(r'subscribe/', ResponseForSubscribe, name='subscribe')
```

```
]
```

```
from django.http import HttpResponse
```

```
from django.shortcuts import render
```

```
from .forms import DetectionForm, Results
```

```
from .prediction_processing import make_prediction
```

```
from datetime import datetime
```

```
from ...tools.logging_config import logger
```

```
from django.shortcuts import redirect
```

```
from ..subscribe.models import User
```

```
def ResponseForDetection(request):
```

```
    if request.method == 'POST':
```

```
        form = DetectionForm(request.POST, request.FILES)
```

```
        if form.is_valid():
```

```
            first_name = form.data['first_name']
```

```
            last_name = form.data['last_name']
```

```
            email = form.data['email']
```

```
        try:
```

```
            user = User.objects.get(first_name=first_name,  
                                   last_name=last_name,  
                                   email=email)
```

```
            logger.info('AUTH: {}'.format(user))
```

```
            form.created = datetime.now()
```

```
            form.save()
```

```
            raw_detection_form = DetectionForm()
```

```
            result, problem = make_prediction(request)
```

```

        logger.info('Result for {}: {}'.format(problem, result))
        result_form = Results.objects.create(email=email, created=datetime.now(),
result=result)
        result_form.save()

        return render(request, "results_detection.html", {'form': raw_detection_form,
                                                             'result': result,
                                                             'problem': problem
                                                             })
    except User.DoesNotExist:
        return redirect('/subscribe/')
    else:
        return HttpResponse('image upload failed with errors: {}'.format(form.errors))
    else:
        form = DetectionForm()
        return render(request, "detect.html", {'form': form})

from django.contrib import admin

from django.apps import AppConfig

class DepartmentsConfig(AppConfig):
    name = 'medical_cases_recognition.apps.departments'

from django.db import models

from django.conf.urls import url
from .views import ResponseForDepartments

urlpatterns = [
    url(r'^$', ResponseForDepartments, name='departments')
]

from django.shortcuts import render

def ResponseForDepartments(request):
    return render(request, "departments.html", locals())

<!doctype html>
<html lang="en">

```

```

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
  <link rel="icon" href="../../static/img/favicon.png" type="image/png">
  <title>Department</title>
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="../../static/css/bootstrap.css">
  <link rel="stylesheet" href="../../static/vendors/linericon/style.css">
  <link rel="stylesheet" href="../../static/css/font-awesome.min.css">
  <link rel="stylesheet" href="../../static/vendors/nice-select/css/nice-select.css">
  <link rel="stylesheet" href="../../static/vendors/animate-css/animate.css">
  <link rel="stylesheet" href="../../static/vendors/jquery-ui/jquery-ui.css">
  <!-- main css -->
  <link rel="stylesheet" href="../../static/css/style.css">
  <link rel="stylesheet" href="../../static/css/responsive.css">
</head>

<body>

  <!--=====Header Menu Area =====-->
  <header class="header_area">
    <div class="main_menu">
      <nav class="navbar navbar-expand-lg navbar-light">
        <div class="container">
          <!-- Brand and toggle get grouped for better mobile display -->
          <a class="navbar-brand logo_h" href="index.html">
            
          </a>
          <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
          <!-- Collect the nav links, forms, and other content for toggling -->
          <div class="collapse navbar-collapse offset"
id="navbarSupportedContent">
            <div class="row ml-0 w-100">
              <div class="col-lg-12 pr-0">
                <ul class="nav navbar-nav center_nav pull-right">
                  <li class="nav-item">

```

```

        <a class="nav-link" href="{% url 'home' %}">Home</a>
    </li>
    <li class="nav-item active">
        <a class="nav-link" href="{% url 'departments' %}">Services</a>
    </li>
</ul>
</div>
</div>
</div>
</div>
</div>
</nav>
</div>
</header>
<!--=====Header Menu Area =====-->
<!--=====Banner Area =====>
<section class="banner_area">
    <div class="banner_inner d-flex align-items-center">
        <div class="container">
            <div class="banner_content text-left">
                <h2>Procedure Category</h2>
                <div class="page_link">
                    <a href="{% url 'home' %}">Home</a>
                    <a href="{% url 'departments' %}">Services</a>
                </div>
            </div>
        </div>
    </div>
</section>
<!--=====End Banner Area =====>

<!--===== Procedure Category Area =====>
>
<section class="procedure_category section_gap_custom">
    <div class="container">
        <div class="row justify-content-center section-title-wrap">
            <div class="col-lg-12">
                <h1>Procedure Category</h1>
                <p>
                    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
                    tempor incididunt ut labore et dolore magna aliqua.
                </p>
            </div>
        </div>
    </div>
    <div class="row">

```



```

<div class="col-lg-4 mb-30">
  <div class="categories_post">
    
    <div class="categories_details">
      <div class="categories_text">
        <div class="border_line"></div>
        <a href="{ % url 'detection' % }">
          <h5>Pneumonia detection via Chest X-Ray</h5>
        </a>
        <div class="border_line"></div>
      </div>
    </div>
  </div>
</div>
<div class="col-lg-4 mb-30">
  <div class="categories_post">
    
    <div class="categories_details">
      <div class="categories_text">
        <div class="border_line"></div>
        <a href="{ % url 'detection' % }">
          <h5>Diabetic Retinopathy Detection</h5>
        </a>
        <div class="border_line"></div>
      </div>
    </div>
  </div>
</div>
<div class="col-lg-4 mb-30">
  <div class="categories_post">
    
    <div class="categories_details">
      <div class="categories_text">
        <div class="border_line"></div>
        <a href="{ % url 'detection' % }">
          <h5>Intracranial Hemorrhage Detection</h5>
        </a>
        <div class="border_line"></div>
      </div>
    </div>
  </div>
</div>

```

```

    </div>
  </div>
</div>
</section>
<!--===== End Procedure Category Area
=====-->
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="../../static/js/jquery-3.2.1.min.js"></script>
<script src="../../static/js/popper.js"></script>
<script src="../../static/js/bootstrap.min.js"></script>
<script src="../../static/vendors/nice-select/js/jquery.nice-select.min.js"></script>
<script src="../../static/js/jquery.ajaxchimp.min.js"></script>
<script src="../../static/https://cdnjs.cloudflare.com/ajax/libs/Counter-
Up/1.0.0/jquery.counterup.min.js"></script>
<script
src="../../static/https://cdnjs.cloudflare.com/ajax/libs/waypoints/4.0.1/jquery.waypoin
ts.min.js"></script>
<script src="../../static/js/mail-script.js"></script>
<script src="../../static/js/custom.js"></script>
</body>

</html>

<!doctype html>
<html lang="en">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
  <link rel="icon" href="../../static/img/favicon.png" type="image/png">
  <title>Hospice Medical</title>
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="../../static/css/bootstrap.css">
  <link rel="stylesheet" href="../../static/vendors/linericon/style.css">
  <link rel="stylesheet" href="../../static/css/font-awesome.min.css">
  <link rel="stylesheet" href="../../static/vendors/owl-
carousel/owl.carousel.min.css">
  <link rel="stylesheet" href="../../static/vendors/lightbox/simpleLightbox.css">
  <link rel="stylesheet" href="../../static/vendors/nice-select/css/nice-select.css">
  <link rel="stylesheet" href="../../static/vendors/animate-css/animate.css">
  <link rel="stylesheet" href="../../static/vendors/jquery-ui/jquery-ui.css">
  <!-- main css -->

```

```

<link rel="stylesheet" href="../../static/css/style.css">
<link rel="stylesheet" href="../../static/css/responsive.css">
</head>

<body>

<!--=====Header Menu Area =====-->
<header class="header_area">
  <div class="main_menu">
    <nav class="navbar navbar-expand-lg navbar-light">
      <div class="container">
        <!-- Brand and toggle get grouped for better mobile display -->
        <a class="navbar-brand logo_h" href="index.html">
          
        </a>
        <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse offset"
id="navbarSupportedContent">
          <div class="row ml-0 w-100">
            <div class="col-lg-12 pr-0">
              <ul class="nav navbar-nav center_nav pull-right">
                <li class="nav-item">
                  <a class="nav-link" href="{% url 'home' %}">Home</a>
                </li>
                <li class="nav-item active">
                  <a class="nav-link" href="{% url 'departments' %}">Services</a>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </div>
    </nav>
  </div>
</header>

<!--=====Header Menu Area =====-->
<!--=====Banner Area =====-->

```

```

<section class="banner_area">
  <div class="banner_inner d-flex align-items-center">
    <div class="container">
      <div class="banner_content text-left">
        <h2>Detection Page</h2>
        <div class="page_link">
          <a href="{% url 'home' %}">Home</a>
          <a href="{% url 'detection' %}">Detection</a>
        </div>
      </div>
    </div>
  </div>
</section>
<!--=====End Banner Area =====-->
<!--=====Contact Area =====-->
<section class="contact_area p_120">
  <div class="container">
    <div class="row">
      <div class="col-lg-9">
        <form method="post" enctype="multipart/form-data">
          {{ form.as_p }}
          {% csrf_token %}
          <div class="col-md-12 text-right">
            <button type="submit" value="submit" class="btn submit_btn">Send Request</button>
          </div>
        </form>
      </div>
    </div>
  </div>
</section>
<!--=====Contact Area =====-->
<!--=====Contact Success and Error message Area =====-->
<div id="success" class="modal modal-message fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <i class="fa fa-close"></i>
        </button>
      <div class="modal-body">
        <h2>Thank you</h2>
        <p>Your request is successfully sent...</p>
      </div>
    </div>
  </div>
</div>

```

```

    </div>
  </div>
</div>

<!-- Modals error -->

<div id="error" class="modal modal-message fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
          <i class="fa fa-close"></i>
        </button>
        <h2>Sorry !</h2>
        <p> Something went wrong </p>
      </div>
    </div>
  </div>
</div>

<!--=====End Contact Success and Error message Area
=====-->

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="../../static/js/jquery-3.2.1.min.js"></script>
<script src="../../static/js/popper.js"></script>
<script src="../../static/js/bootstrap.min.js"></script>
<!-- <script src="vendors/lightbox/simpleLightbox.min.js"></script> -->
<script src="../../static/vendors/nice-select/js/jquery.nice-select.min.js"></script>
<!-- <script src="vendors/isotope/imagesloaded.pkgd.min.js"></script> -->
<script src="../../static/js/jquery.ajaxchimp.min.js"></script>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyA7nx22ZmINyK9TGIXD
EXGVxghC43Ox6qA"></script>
<script src="../../static/https://cdnjs.cloudflare.com/ajax/libs/Counter-
Up/1.0.0/jquery.counterup.min.js"></script>
<script
src="../../static/https://cdnjs.cloudflare.com/ajax/libs/waypoints/4.0.1/jquery.waypoin
ts.min.js"></script>
<script src="../../static/js/mail-script.js"></script>
<script src="../../static/js/custom.js"></script>
</body>

</html>

```

```

<!doctype html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
  <link rel="icon" href="../../static/img/favicon.png" type="image/png">
  <title>MedicalAI</title>
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="../../static/css/bootstrap.css">
  <link rel="stylesheet" href="../../static/vendors/linericon/style.css">
  <link rel="stylesheet" href="../../static/css/font-awesome.min.css">
  <link rel="stylesheet" href="../../static/vendors/owl-
carousel/owl.carousel.min.css">
  <link rel="stylesheet" href="../../static/vendors/lightbox/simpleLightbox.css">
  <link rel="stylesheet" href="../../static/vendors/nice-select/css/nice-select.css">
  <link rel="stylesheet" href="../../static/vendors/animate-css/animate.css">
  <link rel="stylesheet" href="../../static/vendors/jquery-ui/jquery-ui.css">
  <!-- main css -->
  <link rel="stylesheet" href="../../static/css/style.css">
  <link rel="stylesheet" href="../../static/css/responsive.css">
</head>

<body>

  <!--=====Header Menu Area =====-->
  <header class="header_area">
    <div class="main_menu">
      <nav class="navbar navbar-expand-lg navbar-light">
        <div class="container">
          <!-- Brand and toggle get grouped for better mobile display -->
          <a class="navbar-brand logo_h" href="index.html">
            
          </a>
          <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
          <!-- Collect the nav links, forms, and other content for toggling -->
          <div class="collapse navbar-collapse offset"

```



```

Up/1.0.0/jquery.counterup.min.js"></script>
<script
src="../../static/https://cdnjs.cloudflare.com/ajax/libs/waypoints/4.0.1/jquery.waypoin
ts.min.js"></script>
<script src="../../static/js/mail-script.js"></script>
<script src="../../static/js/custom.js"></script>
</body>

</html>

<!doctype html>
<html lang="en">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
  <link rel="icon" href="../../static/img/favicon.png" type="image/png">
  <title>Hospice Medical</title>
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="../../static/css/bootstrap.css">
  <link rel="stylesheet" href="../../static/vendors/linericon/style.css">
  <link rel="stylesheet" href="../../static/css/font-awesome.min.css">
  <link rel="stylesheet" href="../../static/vendors/owl-
carousel/owl.carousel.min.css">
  <link rel="stylesheet" href="../../static/vendors/lightbox/simpleLightbox.css">
  <link rel="stylesheet" href="../../static/vendors/nice-select/css/nice-select.css">
  <link rel="stylesheet" href="../../static/vendors/animate-css/animate.css">
  <link rel="stylesheet" href="../../static/vendors/jquery-ui/jquery-ui.css">
  <!-- main css -->
  <link rel="stylesheet" href="../../static/css/style.css">
  <link rel="stylesheet" href="../../static/css/responsive.css">
</head>

<body>

  <!--=====Header Menu Area =====-->
  <header class="header_area">
    <div class="main_menu">
      <nav class="navbar navbar-expand-lg navbar-light">
        <div class="container">
          <!-- Brand and toggle get grouped for better mobile display -->
          <a class="navbar-brand logo_h" href="index.html">
            <img src="../../static/img/new_logo.jpg" alt="", width="150",

```



```

height="110">
    </a>
    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
    </button>
    <!-- Collect the nav links, forms, and other content for toggling -->
    <div class="collapse navbar-collapse offset"
id="navbarSupportedContent">
        <div class="row ml-0 w-100">
            <div class="col-lg-12 pr-0">
                <ul class="nav navbar-nav center_nav pull-right">
                    <li class="nav-item">
                        <a class="nav-link" href="{% url 'home' %}">Home</a>
                    </li>
                    <li class="nav-item active">
                        <a class="nav-link" href="{% url 'departments' %}">Services</a>
                    </li>
                </ul>
            </div>
        </div>
    </div>
</nav>
</div>
</header>
<!--=====Header Menu Area =====-->

<!--===== Banner Area =====-->
<section class="banner_area">
    <div class="banner_inner d-flex align-items-center">
        <div class="container">
            <div class="banner_content text-left">
                <h2>Detection Page</h2>
                <div class="page_link">
                    <a href="{% url 'home' %}">Home</a>
                    <a href="{% url 'detection' %}">Detection</a>
                </div>
            </div>
        </div>
    </div>
</section>

```

```
<!--=====End Banner Area =====-->
```

```
<!--=====Contact Area =====-->
```

```
<section class="contact_area p_120">
  <div class="container">
    <div class="row">
      <div class="col-lg-9">
        {% if problem == 'Pneumonia' %}
          
          <div style="clear: left">
            <h3>Examination revealed that the patient have {{ result }}.</h3>
            {% if result == 'Pneumonia'%}
            <h3>Contact your doctor immediately.</h3>
            {% else %}
            <h3>Have a nice day!</h3>
            {% endif %}
          </div>
          {% endif %}
          {% if problem == 'Diabetic' %}
          
          <div style="clear: left">
            <h3>Examination revealed that the patient have {{ result }}.</h3>
            {% if result != 'No DR'%}
            <h3>Contact your doctor immediately.</h3>
            {% else %}
            <h3>Have a nice day!</h3>
            {% endif %}
          </div>
          {% endif %}
          {% if problem == 'Hemorrhage' %}
          
          <div style="clear: left">
            <h3>Examination revealed that the patient have {{ result }}.</h3>
            {% if result != 'No Hemorrhage' %}
            <h3>Contact your doctor immediately.</h3>
            {% else %}
            <h3>Have a nice day!</h3>
            {% endif %}
          </div>
          {% endif %}
        </div>
      </div>
    </div>
  </div>
</div>
```

```

</div>
</section>
<!--=====Contact Area =====-->

<!--=====Contact Success and Error message Area
=====-->
<div id="success" class="modal modal-message fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
          <i class="fa fa-close"></i>
        </button>
        <h2>Thank you</h2>
        <p>Your request is successfully sent...</p>
      </div>
    </div>
  </div>
</div>

<!-- Modals error -->

<div id="error" class="modal modal-message fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
          <i class="fa fa-close"></i>
        </button>
        <h2>Sorry !</h2>
        <p>Something went wrong </p>
      </div>
    </div>
  </div>
</div>

<!--=====End Contact Success and Error message Area
=====-->

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="../../static/js/jquery-3.2.1.min.js"></script>
<script src="../../static/js/popper.js"></script>
<script src="../../static/js/bootstrap.min.js"></script>

```

```

<!-- <script src="vendors/lightbox/simpleLightbox.min.js"></script> -->
<script src="../../static/vendors/nice-select/js/jquery.nice-select.min.js"></script>
<!-- <script src="vendors/isotope/imagesloaded.pkgd.min.js"></script> -->
<script src="../../static/js/jquery.ajaxchimp.min.js"></script>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyA7nx22ZmINyk9TGjXD
EXGVxghC43Ox6qA"></script>
<script src="../../static/https://cdnjs.cloudflare.com/ajax/libs/Counter-
Up/1.0.0/jquery.counterup.min.js"></script>
<script
src="../../static/https://cdnjs.cloudflare.com/ajax/libs/waypoints/4.0.1/jquery.waypoin
ts.min.js"></script>
<script src="../../static/js/mail-script.js"></script>
<script src="../../static/js/custom.js"></script>
</body>

</html>

<!doctype html>
<html lang="en">

<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
<link rel="icon" href="../../static/img/favicon.png" type="image/png">
<title>Services</title>
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="../../static/css/bootstrap.css">
<link rel="stylesheet" href="../../static/vendors/linericon/style.css">
<link rel="stylesheet" href="../../static/css/font-awesome.min.css">
<link rel="stylesheet" href="../../static/vendors/owl-
carousel/owl.carousel.min.css">
<link rel="stylesheet" href="../../static/vendors/lightbox/simpleLightbox.css">
<link rel="stylesheet" href="../../static/vendors/nice-select/css/nice-select.css">
<link rel="stylesheet" href="../../static/vendors/animate-css/animate.css">
<link rel="stylesheet" href="../../static/vendors/jquery-ui/jquery-ui.css">
<!-- main css -->
<link rel="stylesheet" href="../../static/css/style.css">
<link rel="stylesheet" href="../../static/css/responsive.css">
</head>

<body>

```

```

<!--=====Header Menu Area =====-->
<header class="header_area">
  <div class="main_menu">
    <nav class="navbar navbar-expand-lg navbar-light">
      <div class="container">
        <!-- Brand and toggle get grouped for better mobile display -->
        <a class="navbar-brand logo_h" href="index.html">
          
        </a>
        <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse offset"
id="navbarSupportedContent">
          <div class="row ml-0 w-100">
            <div class="col-lg-12 pr-0">
              <ul class="nav navbar-nav center_nav pull-right">
                <li class="nav-item">
                  <a class="nav-link" href="{% url 'home' %}">Home</a>
                </li>
                <li class="nav-item ">
                  <a class="nav-link" href="{% url 'departments'
%}">Departments</a>
                </li>
                <li class="nav-item ">
                  <a class="nav-link" href="{% url 'services' %}">Services</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="detect.html">Contact</a>
                </li>
              </ul>
            </div>
          </div>
        </div>
      </nav>
    </div>
  </header>
<!--=====Header Menu Area =====-->
<!--=====Banner Area =====-->

```

```

<section class="banner_area">
  <div class="banner_inner d-flex align-items-center">
    <div class="container">
      <div class="banner_content text-left">
        <h2>Offered Services</h2>
        <div class="page_link">
          <a href="{ % url 'home' % }">Home</a>
          <a href="{ % url 'services' % }">Services</a>
        </div>
      </div>
    </div>
  </div>
</section>
<!--=====End Banner Area =====-->

```

```

<!-- Start Offered Services Area -->

```

```

<section class="service_area section_gap">
  <div class="container">
    <div class="row justify-content-center section-title-wrap">
      <div class="col-lg-12">
        <h1>Our Offered Services</h1>
        <p>
          Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
          tempor incididunt ut labore et dolore magna aliqua.
        </p>
      </div>
    </div>
  </div>

```

```

<div class="row">
  <div class="col-lg-3 col-md-6 mb-60">
    <div class="single_service">
      <span class="lnr lnr-rocket"></span>
      <a href="#">
        <h4>24/7 Emergency</h4>
      </a>
      <p>

```

inappropriate behavior is often laughed off as “boys will be boys,” women face higher conduct women face higher conduct.

```

      </p>
    </div>
  </div>
  <div class="col-lg-3 col-md-6 mb-60">
    <div class="single_service">
      <span class="lnr lnr-heart-pulse"></span>
      <a href="#">

```

Expert Consultation

inappropriate behavior is often laughed off as “boys will be boys,” women face higher conduct women face higher conduct.

Intensive Care

inappropriate behavior is often laughed off as “boys will be boys,” women face higher conduct women face higher conduct.

Family Planning

inappropriate behavior is often laughed off as “boys will be boys,” women face higher conduct women face higher conduct.

24/7 Emergency

inappropriate behavior is often laughed off as “boys will be boys,” women face higher conduct women face higher conduct.


```

<script src="../../static/vendors/isotope/isotope-min.js"></script>
<script src="../../static/vendors/owl-carousel/owl.carousel.min.js"></script>
<script src="../../static/js/jquery.ajaxchimp.min.js"></script>
<script src="../../static/https://cdnjs.cloudflare.com/ajax/libs/Counter-
Up/1.0.0/jquery.counterup.min.js"></script>
<script
src="../../static/https://cdnjs.cloudflare.com/ajax/libs/waypoints/4.0.1/jquery.waypoin
ts.min.js"></script>
<script src="../../static/js/mail-script.js"></script>
<script src="../../static/js/custom.js"></script>
</body>

</html>

<!doctype html>
<html lang="en">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
  <link rel="icon" href="../../static/img/favicon.png" type="image/png">
  <title>Hospice Medical</title>
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="../../static/css/bootstrap.css">
  <link rel="stylesheet" href="../../static/vendors/linericon/style.css">
  <link rel="stylesheet" href="../../static/css/font-awesome.min.css">
  <link rel="stylesheet" href="../../static/vendors/owl-
carousel/owl.carousel.min.css">
  <link rel="stylesheet" href="../../static/vendors/lightbox/simpleLightbox.css">
  <link rel="stylesheet" href="../../static/vendors/nice-select/css/nice-select.css">
  <link rel="stylesheet" href="../../static/vendors/animate-css/animate.css">
  <link rel="stylesheet" href="../../static/vendors/jquery-ui/jquery-ui.css">
  <!-- main css -->
  <link rel="stylesheet" href="../../static/css/style.css">
  <link rel="stylesheet" href="../../static/css/responsive.css">
</head>

<body>

  <!--=====Header Menu Area =====-->
  <header class="header_area">
    <div class="main_menu">
      <nav class="navbar navbar-expand-lg navbar-light">

```

```

<div class="container">
  <!-- Brand and toggle get grouped for better mobile display -->
  <a class="navbar-brand logo_h" href="index.html">
    
  </a>
  <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
  </button>
  <!-- Collect the nav links, forms, and other content for toggling -->
  <div class="collapse navbar-collapse offset"
id="navbarSupportedContent">
    <div class="row ml-0 w-100">
      <div class="col-lg-12 pr-0">
        <ul class="nav navbar-nav center_nav pull-right">
          <li class="nav-item">
            <a class="nav-link" href="{% url 'home' %}">Home</a>
          </li>
          <li class="nav-item active">
            <a class="nav-link" href="{% url 'departments' %}">Services</a>
          </li>
        </ul>
      </div>
    </div>
  </div>
</nav>
</div>
</header>
<!--=====Header Menu Area =====-->

<!--===== Banner Area =====-->
<section class="banner_area">
  <div class="banner_inner d-flex align-items-center">
    <div class="container">
      <div class="banner_content text-left">
        <h2>Detection Page</h2>
        <div class="page_link">
          <a href="{% url 'home' %}">Home</a>
          <a href="{% url 'subscribe' %}">Subscribe</a>
        </div>
      </div>
    </div>
  </div>
</section>

```

```

    </div>
  </div>
</div>
</section>
<!--=====End Banner Area =====-->

<!--=====Contact Area =====-->
<section class="contact_area p_120">
  <div class="container">
    <div class="row">
      <div class="col-lg-9">
        <form method="post" enctype="multipart/form-data">
          {{ user_form.as_p }}
          {{ general_form.as_p }}
          {% csrf_token %}
          <div class="col-md-12 text-right">
            <button type="submit" value="submit" class="btn
submit_btn">Send Request</button>
          </div>
        </form>
      </div>
    </div>
  </div>
</section>
<!--=====Contact Area =====-->

<!--=====Contact Success and Error message Area
=====-->
<div id="success" class="modal modal-message fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
          <i class="fa fa-close"></i>
        </button>
        <h2>Thank you</h2>
        <p>Your request is successfully sent...</p>
      </div>
    </div>
  </div>
</div>

<!-- Modals error -->

```

```

<div id="error" class="modal modal-message fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
          <i class="fa fa-close"></i>
        </button>
        <h2>Sorry !</h2>
        <p> Something went wrong </p>
      </div>
    </div>
  </div>
</div>

<!--=====End Contact Success and Error message Area
=====-->

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="../../static/js/jquery-3.2.1.min.js"></script>
<script src="../../static/js/popper.js"></script>
<script src="../../static/js/bootstrap.min.js"></script>
<!-- <script src="vendors/lightbox/simpleLightbox.min.js"></script> -->
<script src="../../static/vendors/nice-select/js/jquery.nice-select.min.js"></script>
<!-- <script src="vendors/isotope/imagesloaded.pkgd.min.js"></script> -->
<script src="../../static/js/jquery.ajaxchimp.min.js"></script>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyA7nx22ZmINyK9TGIXD
EXGVxghC43Ox6qA"></script>
<script src="../../static/https://cdnjs.cloudflare.com/ajax/libs/Counter-
Up/1.0.0/jquery.counterup.min.js"></script>
<script
src="../../static/https://cdnjs.cloudflare.com/ajax/libs/waypoints/4.0.1/jquery.waypoin
ts.min.js"></script>
<script src="../../static/js/mail-script.js"></script>
<script src="../../static/js/custom.js"></script>
</body>

</html>

<!doctype html>
<html lang="en">

<head>
  <!-- Required meta tags -->

```

```

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
<link rel="icon" href="../../static/img/favicon.png" type="image/png">
<title>Hospice Medical</title>
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="../../static/css/bootstrap.css">
<link rel="stylesheet" href="../../static/vendors/linericon/style.css">
<link rel="stylesheet" href="../../static/css/font-awesome.min.css">
<link rel="stylesheet" href="../../static/vendors/owl-
carousel/owl.carousel.min.css">
<link rel="stylesheet" href="../../static/vendors/lightbox/simpleLightbox.css">
<link rel="stylesheet" href="../../static/vendors/nice-select/css/nice-select.css">
<link rel="stylesheet" href="../../static/vendors/animate-css/animate.css">
<link rel="stylesheet" href="../../static/vendors/jquery-ui/jquery-ui.css">
<!-- main css -->
<link rel="stylesheet" href="../../static/css/style.css">
<link rel="stylesheet" href="../../static/css/responsive.css">
</head>

<body>

<!--=====Header Menu Area =====-->
<header class="header_area">
  <div class="main_menu">
    <nav class="navbar navbar-expand-lg navbar-light">
      <div class="container">
        <!-- Brand and toggle get grouped for better mobile display -->
        <a class="navbar-brand logo_h" href="index.html">
          
        </a>
        <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse offset"
id="navbarSupportedContent">
          <div class="row ml-0 w-100">
            <div class="col-lg-12 pr-0">
              <ul class="nav navbar-nav center_nav pull-right">

```

```

        <li class="nav-item">
            <a class="nav-link" href="{% url 'home' %}">Home</a>
        </li>
        <li class="nav-item active">
            <a class="nav-link" href="{% url 'departments' %}">Services</a>
        </li>
    </ul>
</div>
</div>
</div>
</div>
</div>
</nav>
</div>
</header>
<!--=====Header Menu Area =====-->

<!--===== Banner Area =====-->
<section class="banner_area">
    <div class="banner_inner d-flex align-items-center">
        <div class="container">
            <div class="banner_content text-left">
                <h2>Detection Page</h2>
                <div class="page_link">
                    <a href="{% url 'home' %}">Home</a>
                    <a href="{% url 'subscribe' %}">Subscribe</a>
                </div>
            </div>
        </div>
    </div>
</div>
</section>
<!--=====End Banner Area =====-->

<!--=====Contact Success and Error message Area
=====-->
<div id="success" class="modal modal-message fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h2>Thank you</h2>
                <p>Your request is successfully sent. Now you can view the results</p>
            </div>
        </div>
    </div>
</div>

```

```

<!-- Modals error -->

<div id="error" class="modal modal-message fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
          <i class="fa fa-close"></i>
        </button>
        <h2>Sorry !</h2>
        <p>Something went wrong </p>
      </div>
    </div>
  </div>
</div>

<!--=====End Contact Success and Error message Area
=====-->

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="../../static/js/jquery-3.2.1.min.js"></script>
<script src="../../static/js/popper.js"></script>
<script src="../../static/js/bootstrap.min.js"></script>
<!-- <script src="vendors/lightbox/simpleLightbox.min.js"></script> -->
<script src="../../static/vendors/nice-select/js/jquery.nice-select.min.js"></script>
<!-- <script src="vendors/isotope/imagesloaded.pkgd.min.js"></script> -->
<script src="../../static/js/jquery.ajaxchimp.min.js"></script>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyA7nx22ZmINyK9TGIXD
EXGVxghC43Ox6qA"></script>
<script src="../../static/https://cdnjs.cloudflare.com/ajax/libs/Counter-
Up/1.0.0/jquery.counterup.min.js"></script>
<script
src="../../static/https://cdnjs.cloudflare.com/ajax/libs/waypoints/4.0.1/jquery.waypoin
ts.min.js"></script>
<script src="../../static/js/mail-script.js"></script>
<script src="../../static/js/custom.js"></script>
</body>

</html>

```

Реалізація файлів для налаштування роботи сервісу

import os

```

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
SECRET_KEY = 'lza^z@r&67ur4q_gw9n(wh1szyf#%9)d0xk=lpa_5i!3)5ub2z'
DEBUG = True

```

```

ALLOWED_HOSTS = []

```

```

# Application definition

```

```

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'medical_cases_recognition.apps.home',
    'medical_cases_recognition.apps.departments',
    'medical_cases_recognition.apps.detection',
    'medical_cases_recognition.apps.subscribe'
]

```

```

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

```

```

#AUTH_USER_MODEL = "subscribe.Subscriber"

```

```

ROOT_URLCONF = 'medical_cases_recognition.urls'

```

```

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'medical_cases_recognition/apps/templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',

```



```

        'django.template.context_processors.request',
        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',
    ],
},
],

WSGI_APPLICATION = 'medical_cases_recognition.wsgi.application'

# Database

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}

# Password validation

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
        'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
        'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
        'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

```

```
USE_TZ = True
```

```
# Static files (CSS, JavaScript, Images)
```

```
PROJECT_ROOT = os.path.dirname(os.path.abspath(__file__))
```

```
# STATIC_ROOT = os.path.join(PROJECT_ROOT, 'static')
```

```
STATIC_URL = '/static/'
```

```
STATICFILES_DIRS = (
    os.path.join(PROJECT_ROOT, 'static'),
)
```

```
MEDIA_ROOT = os.path.join(PROJECT_ROOT, 'media')
```

```
MEDIA_URL = '/media/'
```

```
STATICFILES_FINDERS = (
```

```
    'django.contrib.staticfiles.finders.FileSystemFinder',
```

```
    'django.contrib.staticfiles.finders.AppDirectoriesFinder',
```

```
)
```

```
"""medical_cases_recognition URL Configuration"""
```

```
from django.contrib import admin
```

```
from django.conf import settings
```

```
from django.conf.urls import url, include
```

```
from django.conf.urls.static import static
```

```
from django.contrib.staticfiles.urls import staticfiles_urlpatterns
```

```
urlpatterns = [
```

```
    url(r'^admin/', admin.site.urls),
```

```
    url(r'^home/', include('medical_cases_recognition.apps.home.urls')),
```

```
    url(r'^departments/', include('medical_cases_recognition.apps.departments.urls')),
```

```
    url(r'^services/', include('medical_cases_recognition.apps.services.urls')),
```

```
    url(r'^detection/', include('medical_cases_recognition.apps.detection.urls')),
```

```
    url(r'^subscribe/', include('medical_cases_recognition.apps.subscribe.urls'))
```

```
]
```

```
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

```
urlpatterns += staticfiles_urlpatterns()
```